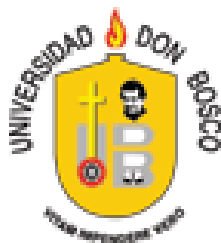


UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA



DISEÑO DE UN COPROCESADOR DIFUSO DE PROPÓSITOS
GENERALES

TRABAJO DE GRADUACIÓN

PRESENTADO POR

Carlos Ernesto Cabrera González

Guillermo Atilio González Penado

PARA OPTAR AL GRADO DE

Ingeniero en Electrónica

Asesor:

Ing. Herbert Edgardo Ascencio Hurtado

Septiembre de 2004

Soyapango – El Salvador – Centro América

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA

AUTORIDADES:

RECTOR
ING. FEDERICO HUGUET RIVERA

VICERRECTOR ACADÉMICO
PBRO. VÍCTOR BERMÚDEZ, sdb

SECRETARIO GENERAL
LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE INGENIERÍA
ING. GODOFREDO GIRÓN

DIRECTOR DE ESCUELA DE ELECTRÓNICA
ING. OSCAR DURÁN VIZCARRA

ASESOR DEL TRABAJO DE GRADUACIÓN
ING. HERBERT EDGARDO ASCENCIO HURTADO

JURADO EVALUADOR
ING. EDGARDO CRUZ ZELEDON
ING. SERGIO MARTIN
ING. EDUARDO RIVERA

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE ELECTRÓNICA

JURADO EVALUADOR DEL TRABAJO DE GRADUACIÓN

Ing. Edgardo Cruz Zeledón
JURADO

Ing. Sergio Martín
JURADO

Ing. Eduardo Rivera
JURADO

Ing. Edgardo Ascencio Hurtado
ASESOR

AGRADECIMIENTOS

A Dios todopoderoso, por haberme permitido llegar hasta este momento, por haberme dado la fuerza y la vida, por haberme dado la esperanza y por haberme dado el consuelo aún en la desesperación; También gracias por habernos permitido concebir un hijo, aun cuando las esperanzas eran pocas.

A mis padres Carlos Gilberto Cabrera y Reina Guadalupe de Cabrera, por haberme dado de su sabiduría, por haberme guiado y enseñado el camino a seguir. Gracias por haberme dado la educación y el estudio, pero principalmente por haberme mostrado el camino de Dios.

A mi esposa Carmen Elena de Cabrera, por haberme apoyado siempre de forma incondicional y principalmente estos últimos meses de ardua labor, agradezco también el haberme dado un Hijo al que aún no conozco pero, quien se, tendrá la belleza y la dulzura de su madre, a quien nunca podré agradecer del todo el haber dado tanta comprensión y lo mejor de si a cambio de nada, gracias por haber estado allí en los momentos de aflicción, en los días de enfermedad, en los días buenos y en los menos buenos, en fin GRACIAS, pero no se como decirlo para que todos entiendan lo que ha significado para mi.

A mi hermana Edna Marisol Cabrera por haberme ayudado tanto como le fue posible e incluso más allá de lo posible, gracias por haber sido mi hermana, de quien he aprendido muchísimo.

A mi abuelita Delfina Argueta, por quererme tanto, por haber cuidado de mi desde que era un bebe, gracias por siempre haber estado preocupada por mi bienestar, también gracias por amar tanto a mi esposa y a mi hijo que esta en camino.

A mis suegros Don Carlos Quintanilla y Doña Beatriz de Quintanilla, por haberme apoyado incondicionalmente casi como un padre a su hijo, gracias por todo el cariño brindado, gracias por haber sufrido conmigo por haber gozado y reído con nosotros.

A mi cuñado Luis Manuel Quintanilla, por haber siempre estado al pendiente de nosotros, del avance de este proyecto, también gracias por esperar con tanta ansia a su sobrino Carlos Andrés.

A la abuelita de mi esposa Doña Ana María Martínez, por haber estado siempre al pendiente de este proyecto, por haberse preocupado, por haber pasado noches de desvelo aún sin haber tenido la obligación de hacerlo, gracias por el apoyo moral brindado.

A Herbert Ascencio, por haber sido un amigo como pocos en esta vida, gracias por todo el tiempo que dedicaste aunque sea solo a escuchar, por estar siempre allí para lo que necesitara, gracias por que amistades tan sinceras y desinteresadas quedan pocas.

A Jorge Ferrarotti, por haber haberme apoyado y confiado en mi siempre, también por haberme mostrado hacia donde debe apuntar mi vida y aspiraciones profesionales, gracias por que sin su consejo este proyecto así como muchas de las experiencias que he adquirido no hubieran sido posibles, así como por haberme enseñado que se puede llegar tan alto como uno desee, a base de empeño, esfuerzo y trabajo. Gracias por la enseñanza de la escuela de la vida, ya que él aprendí las lecciones que no enseñan en ninguna facultad de ninguna universidad.

A Guillermo González, por haber sido un buen compañero todos estos años, con quien hemos pasado una y mil anécdotas, y con quien nos hemos embarcado en decenas de proyectos, algunos con mejor suerte que otros, pero de los que siempre aprendimos a tomar lo mejor.

Al Ing. Héctor Carías, por haber sido siempre tan comprensivo, así mismo por habernos ayudado en los momentos de mayor necesidad, gracias por habernos apoyado durante todos estos años.

Carlos Ernesto Cabrera González

A Dios Todopoderoso por darme la vida y por permitirme terminar este paso importante en mi vida y conservarme las fuerzas para seguir siempre adelante y tratar de ser mejor cada día.

A mi madre, la maestra Milagro Penado por su amor y constante apoyo, por haberme inculcado valores que han permitido mi desarrollo tanto personal como profesional, por haberme enseñado desde pequeño el camino bajo los lineamientos de Dios.

A mi hija Nicole González, por ser la angelita que Dios puso en mi vida para hacerme un mejor ser humano, por darme fuerzas en los momentos que lo requerí y por ser tan divina y cariñosa.

A mis abuelos Maria Higinia y Víctor Manuel, los mejores abuelos del mundo por considerarme uno más de sus hijos y estar siempre a mi lado.

A mis hermanas Pily y Margie, por su infaltable apoyo, amor comprensión y soporte durante toda mi vida y en especial en mis estudios.

A mis tíos Simón Enrique, Freddy y Milagro, por su constante ayuda durante mis estudios.

A mis primos Freddy, Karen y Ana Milagro, más que primos hermanos, sin cuyas palabras y muestras de afecto las cosas no hubiesen sido igual.

A Carlos Ernesto, compañero en tantas batallas en la UDB, unas perdidas otras ganadas pero que juntas fueron forjando el carácter profesional necesario para enfrentar los retos que la vida presenta; por su dedicación en este trabajo, por descifrar mis algoritmos no lineales y en general por haber sido más que un compañero de tesis un amigo a lo largo de la carrera.

A Herberth Ascencio, por su apoyo y amistad a lo largo de mis estudios, por haber aceptado el reto de ser nuestro asesor y por su admirable cualidad de explicar a sus compañeros los temas que el maneja de una manera desinteresada.

A la familia Quintanilla, en especial a Carmen Elena y Carlos Andrés por su total apoyo durante la elaboración de este trabajo y por el tiempo de Carlos Ernesto que cedieron para que finalizáramos la tesis.

A los Ingenieros de Nejapa Power Oscar Palomo, Gerardo Guardado y Walter Polanco, por el apoyo desinteresado brindado para la realización del presente trabajo de graduación.

A mis amigos René Chinchilla y Salvador Choto, quienes durante el tiempo de estudios me brindaron su amistad y me motivaron a seguir estudiando hasta coronar esta carrera.

A mis amigos Hugo Doradea, Alonso Saravia, Herberth Herrera, Alejandro Alvarado, compañeros de trabajo que siempre me instaron a que finalizara esta etapa de mis estudios.

Al Ing. Hector Carías por habernos ayudado en las pruebas realizadas en el CITT y en general por los conocimientos compartidos durante la carrera.

Al Tec. Mario Crespín por su apoyo en el laboratorio de Instrumentación y Control en las pruebas que durante la elaboración del trabajo de graduación fueron realizadas en el CITT.

En general agradezco a toda la gente que de alguna forma u otra aportaron y me ayudaron en mis estudios así como a los docentes y compañeros de la Universidad Don Bosco.

Guillermo Atilio González Penado

INDICE

1. RESEÑA HISTORICA.....	4
1.1 RESEÑA HISTORICA.....	4
1.10 RELACIONES DIFUSAS.....	20
1.11 COMPOSICIÓN DE RELACIONES DIFUSAS.....	21
1.12 REGLAS DIFUSAS.....	22
1.12.1 PROPOSICIONES DIFUSAS.....	23
1.12.2 FUNCION DE IMPLICACIÓN.....	23
1.12.3 FUNCIONES DE IMPLICACIÓN GLOBALES.....	24
1.13 RAZONAMIENTO APROXIMADO.....	26
1.14 MECANISMOS DE INFERENCIA.....	29
1.14.2 PRODUCTO SUMA.....	29
1.15 METODOS DE DEFUZIFICACION.....	30
1.15.1 METODOS CONVENCIONALES.....	30
1.15.2 METODOS SIMPLIFICADOS.....	33
1.16 ESTRUCTURA DE UN SITEMA DIFUSO.....	36
1.17 APLICACIONES INDUSTRIALES.....	37
1.17.1 CONTROL CONVENCIONAL.....	37
1.17.2 CONTROL DIFUSO.....	38
1.18 SOPORTE PARA LOGICA DIFUSA CON EL MCU 68HC12.....	39
1.18.1 Fuzzyficación (MEM).....	40
1.18.2 Evaluación de Reglas (REV y REVW).....	42
1.18.3 Defuzzyficación (WAV).....	48
1.2 DEFINICIÓN.....	5
1.3 CONJUNTOS DIFUSOS.....	8
1.4 VARIABLE LINGÜÍSTICA.....	9
1.5 BASE DE REGLAS.....	10

1.6 MECANISMO DE INFERENCIA.....	11
1.7 FUNCIONES DE PERTENENCIA.....	12
1.8 FUZZY SINGLETON.....	13
1.9 OPERACIONES ENTRE CONJUNTOS DIFUSOS.....	14
1.9.1 UNION.....	14
1.9.2 INTERSECCIÓN.....	15
1.9.3 COMPLEMENTO.....	15
1.9.3.1 LEY DEL TERCIO EXCLUSO.....	16
1.9.3.2 LEY DE LA CONTRADICCIÓN.....	16
1.9.4 INTERSECCION DIFUSA\.....	17
T NORMAS.....	17
1.9.5 UNION DIFUSA\.....	19
S NORMAS.....	19
1.MARCO TEORICO.....	4
2. DEFINICIÓN.....	5
2. INTERFAZ GRAFICA.....	50
2.1 CONJUNTOS DIFUSOS.....	8
2.2 VARIABLE LINGÜÍSTICA.....	9
2.3 BASE DE REGLAS.....	10
2.4 MECANISMO DE INFERENCIA.....	11
2.5 FUNCIONES DE PERTENENCIA.....	12
2.6 FUZZY SINGLETON.....	13
2.7 OPERACIONES ENTRE CONJUNTOS DIFUSOS.....	14
2.7.1 UNION.....	14
2.7.2 INTERSECCIÓN.....	15
2.7.3 COMPLEMENTO.....	15
2.7.4 INTERSECCION DIFUSA\.....	17
T NORMAS.....	17
2.7.5 UNION DIFUSA\.....	19
S NORMAS.....	19
2.8 RELACIONES DIFUSAS.....	20

2.8.1 COMPOSICIÓN DE RELACIONES DIFUSAS.....	21
3. REGLAS DIFUSAS.....	22
3. RESULTADOS OBTENIDOS CON EL CONTROLADOR DIFUSO	
.....	75
3.1 PROPOSICIONES DIFUSAS.....	23
3.2 FUNCION DE IMPLICACIÓN.....	23
3.2.1 FUNCIONES DE IMPLICACIÓN GLOBALES.....	24
4. RAZONAMIENTO APROXIMADO.....	26
4.CODIGO FUENTE DE LA APLICACIÓN DEL SIMULADOR	
DIFUSO EN VISUAL BASIC.....	132
5. MECANISMOS DE INFERENCIA.....	29
5.1 CONTROL DE NIVEL DE LIQUIDO EN UN TANQUE.....	109
5.1 MIN-MAX.....	29
5.2 CONTROL DE VELOCIDAD EN UN MOTOR.....	114, 118
5.2 PRODUCTO SUMA.....	29
5.CODIGO ENSAMBLADOR PARA EL CONTROLADOR DIFUSO	
.....	109
6. MANUAL DEL USUARIO.....	371, 427
6. METODOS DE DEFUZIFICACION.....	30
6.1 METODOS CONVENCIONALES.....	30
Centro de sumas (CoS).....	31
Primer Máximo (FoM).....	32
Ultimo Máximo (LoM).....	32
6.2 METODOS SIMPLIFICADOS.....	33
Media Difusa (FM).....	33
Media Difusa Ponderada (WFM).....	34
Método \“level grading\”.....	34
Método de calidad (QM).....	34
Método de Yager (YM).....	35
7. ESTRUCTURA DE UN SITEMA DIFUSO.....	36
8. APLICACIONES INDUSTRIALES.....	37

8.1 CONTROL CONVENCIONAL.....	37
8.2 CONTROL DIFUSO.....	38
9. SOPORTE PARA LOGICA DIFUSA CON EL MCU 68HC12.....	39
Defuzzyficación (WAV).....	48
Evaluación de Reglas (REV y REVW).....	42
Fuzzyficación (MEM).....	40

1.MARCO TEORICO

1.1 RESEÑA HISTORICA

Como un intento de construir máquinas cuyos modelos computacionales se aproximarán a los mecanismos de razonamiento humanos, surge, a mediados de los años 50, diferentes técnicas de **Inteligencia Artificial**. Los **Sistemas Expertos** tratan de emular nuestra capacidad para estructurar la información. Las **Redes Neuronales** intentan emular la organización de nuestro cerebro [3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].

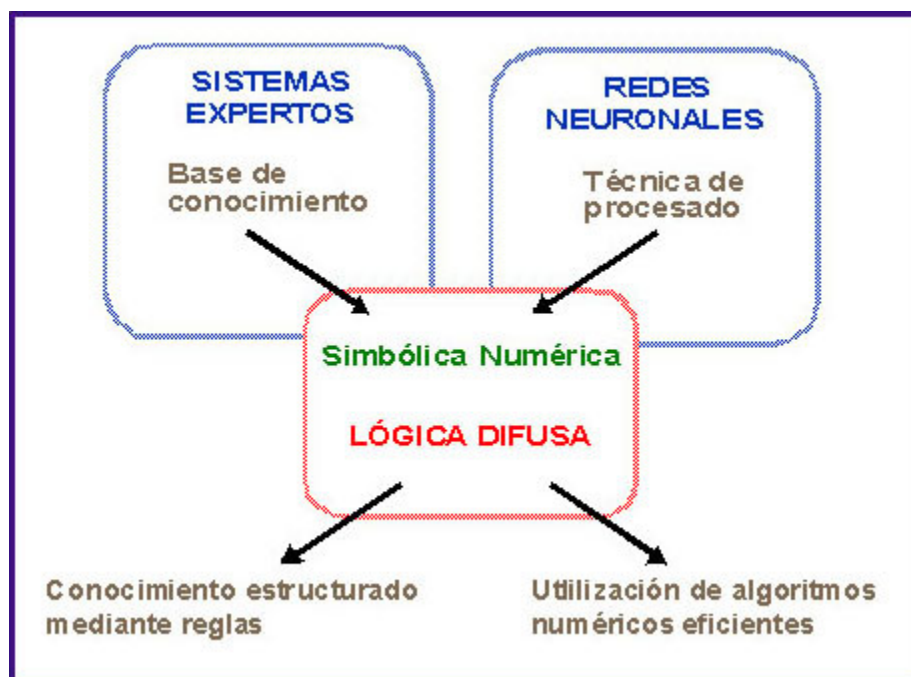


FIGURA 1.1. Relación esquemática entre los sistemas de lógica difusa y los sistemas expertos y redes neuronales

Al igual que los sistemas expertos y las redes neuronales, los sistemas basados en lógica difusa obtienen la salida del sistema en función de sus entradas sin necesidad de recurrir a la utilización de un modelo analítico. Comparten con los primeros la característica de representar el conocimiento de forma estructurada (mediante reglas) y con los segundos la facilidad para manipular la información de forma numérica. Esta última cualidad hace especialmente atractiva la implementación en *hardware* de sistemas de inferencia basados en lógica difusa, representado esto en la figura 1.1.

La **Teoría de conjuntos difusos** desarrollada por Lofti A. Zadeh a mediados de la década de los 60 y las técnicas de inferencia basadas en ella (**Lógica difusa**) proporcionan una herramienta adecuada para modelar la incertidumbre presente en el lenguaje natural y emular los mecanismos de razonamiento aproximado utilizados por el cerebro humano [1.-Inteligencia artificial, una nueva síntesis].

1.2 DEFINICIÓN

El concepto fundamental en que se basa la lógica difusa es el de **conjunto difuso**, un tipo de conjunto caracterizado porque los elementos del universo de discurso en el que está definido pueden pertenecer a él en un cierto grado, representado por una **función de pertenencia**, tal como se presenta en la figura 1.2.

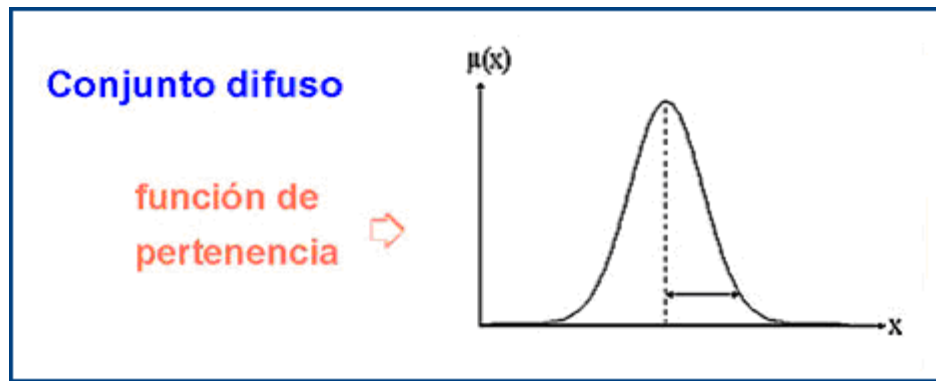


FIGURA 1.2. Representación gráfica de una función de pertenencia para un conjunto difuso

Un sistema difuso es un sistema basado en el conocimiento. Dicho conocimiento se estructura en un conjunto de reglas simbólicas del tipo “*si-entonces*”, que utilizan términos del lenguaje natural para representar información vaga o imprecisa. Por otra parte, desde el punto de vista matemático, un sistema difuso es un sistema determinista que proporciona un mapeado no-lineal entre entradas y salidas. Las dos características anteriores son la base del **control difuso** [3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].

Otra de las características fundamentales de un sistema difuso es su habilidad para manejar ambigüedad. La posibilidad de extraer conclusiones con valores estimados a partir de información incompleta o incierta es explotada en otras muchas aplicaciones de la lógica difusa a diferentes campos de la Ingeniería y las Ciencias Sociales. Entre los diferentes dominios de aplicación cabe destacar los sistemas de **reconocimiento de patrones** (procesado de voz e imágenes, clasificación, etc.), los **sistemas expertos difusos** (para diagnóstico en medicina, análisis financiero, toma de decisiones, etc.) y las **bases de datos difusas** (que procesan información imprecisa y permiten a los usuarios formular preguntas en lenguaje natural); sin duda el campo de la Ingeniería donde la

lógica difusa ha sido aplicada en más ocasiones y con mejores resultados. Ver figura 1.3

[3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].

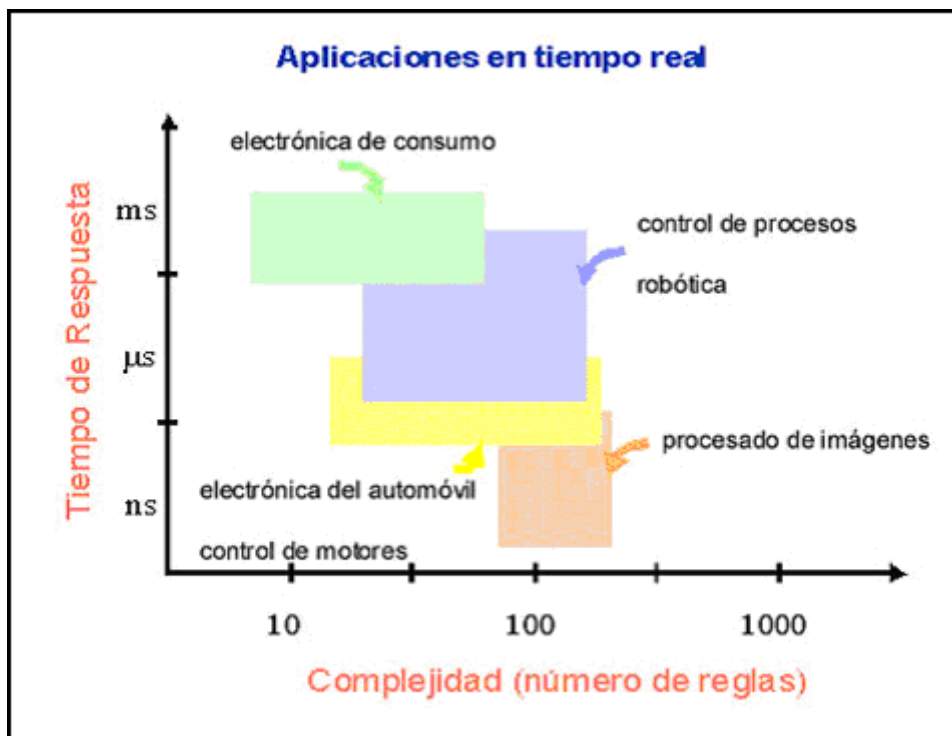


FIGURA 1.3 Ilustración de la complejidad de los sistemas de inferencia y los tiempos de respuesta requeridos en diferentes aplicaciones de la lógica difusa

1.3 CONJUNTOS DIFUSOS

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

El **grado de pertenencia** es un número real dentro del intervalo $[0, 1]$ que indica en qué proporción pertenece un determinado elemento a un conjunto. De este modo, si un elemento tiene un grado de pertenencia '0' respecto a un conjunto dado, será equivalente a decir que dicho elemento no pertenece a dicho conjunto. Análogamente, si un elemento tiene un grado de pertenencia '1', se dirá que dicho elemento se encuentra totalmente dentro del conjunto. Esto se representa en la figura 1.4 [3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].

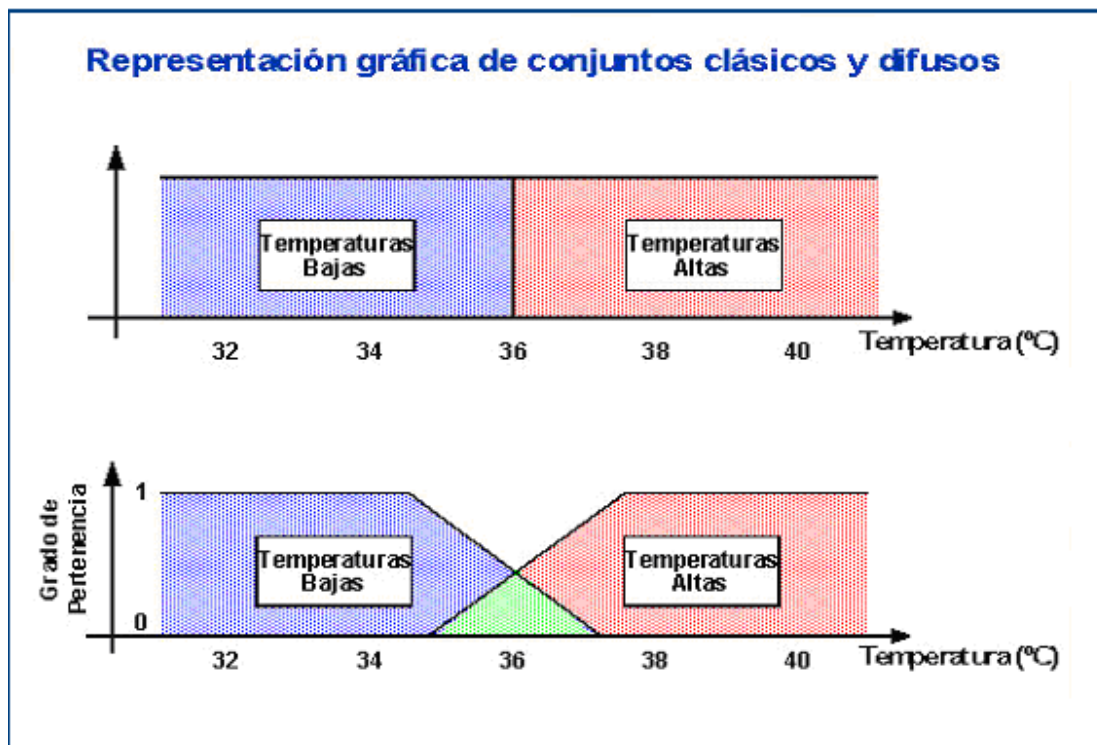


FIGURA 1.4. La ilustración muestra la representación gráfica de los conjuntos Boléanos y Difusos en el mismo orden.

1.4 VARIABLE LINGÜÍSTICA

Una **variable lingüística**, representada en la figura 1.5, es una variable cuyos valores pueden expresarse mediante términos del lenguaje natural. Los diferentes términos o valores lingüísticos se representan mediante conjuntos difusos caracterizados por funciones de pertenencia dentro del universo de discurso.



FIGURA 1.5. Ejemplo de una variable lingüística que se compone de cinco conjuntos difusos en el que resalta el correspondiente a “alta” lo cual conlleva a la premisa que en ese rango de $v_{(t)}$ “la velocidad es alta”

Formalmente una variable lingüística viene caracterizada por n elementos. A

continuación se muestra un ejemplo con cuatro términos para la variable lingüística

‘**Temperatura**’ que son baja, normal, alta y muy alta [3.- FLEB, a Fuzzy logic E-Book,

www.esime.edu.es].

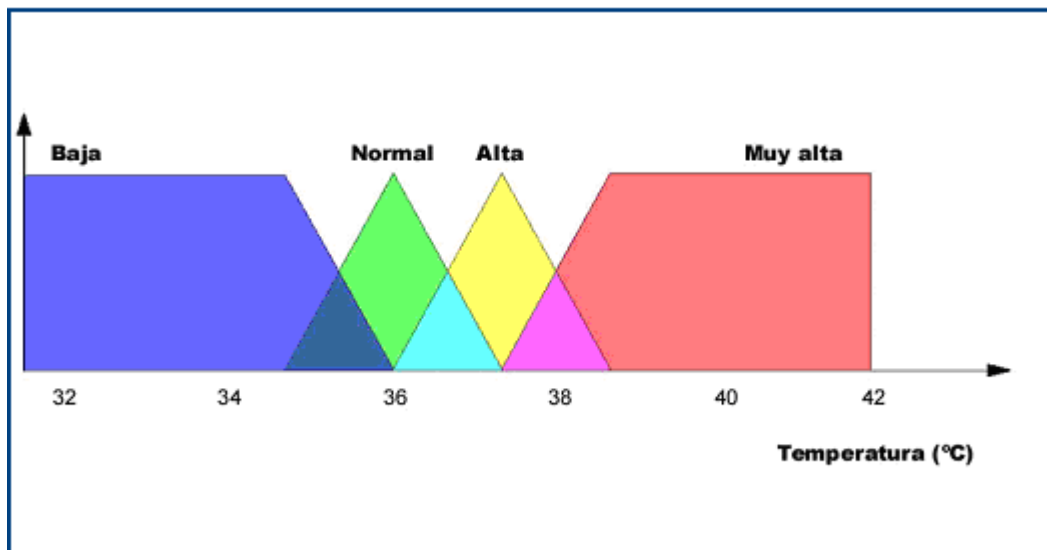


FIGURA 1.6. Representación gráfica de los términos definidos para la variable lingüística temperatura

1.5 BASE DE REGLAS

Un sistema de inferencia basado en lógica difusa, o **sistema difuso**, está formado por un conjunto de las reglas del tipo “*si x es A entonces y es B* ”, donde x e y son las variables del sistema, y A y B son términos lingüísticos como ‘alto’, ‘bajo’, ‘medio’, ‘positivo’, ‘negativo’, etc. Según se muestra en la figura 1.7 [3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].

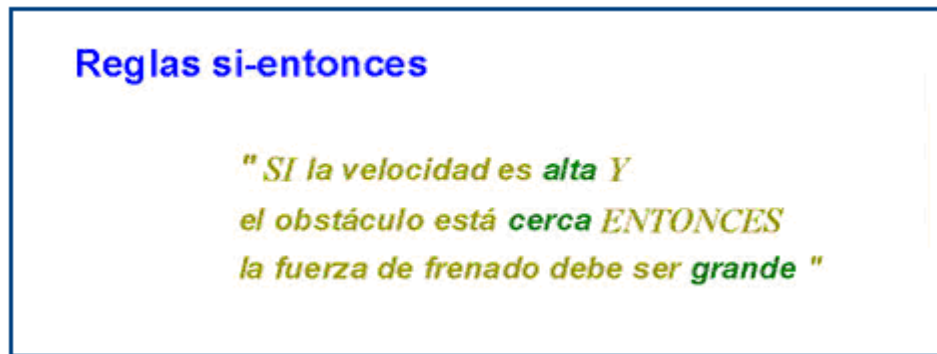


FIGURA 1.7. Ejemplo de una regla “si-entonces”, donde puede verse claramente las premisas antecedentes conectadas entre sí mediante un conectivo Y, de las cuales se deriva la conclusión correspondiente

1.6 MECANISMO DE INFERENCIA

Las técnicas de razonamiento utilizadas en lógica difusa es una generalización de los mecanismos de inferencia empleados en la lógica bivaluada tradicional, pero a diferencia de un sistema experto convencional, en un sistema difuso varias reglas pueden estar activas simultáneamente con diferentes grados de activación. Cuando varía una de sus entradas el sistema evalúa todas las reglas para inferir una conclusión o salida. En la figura 1.8 se presenta los componentes de un mecanismo de inferencia [3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].



FIGURA 1.8. Mecanismo de inferencia que ilustra el pensamiento aproximado que parte de una premisa x es A' lo cual tiene cierto grado de verdad respecto de x es A , por tanto sobre la base de la regla se concluye que y es B' con similar grado de verdad respecto de y es B .

1.7 FUNCIONES DE PERTENENCIA

Desde el punto de vista matemático, un conjunto difuso puede representarse mediante un conjunto ordenado de pares que asignan un grado de pertenencia a cada elemento u del universo de discurso U .

$$F = \{(u, \mu_F(u)) \mid u \in U\} \quad \text{Ec.1}$$

La **función de pertenencia** $\mu_F(u)$ describe, por tanto, el grado de pertenencia de los diferentes elementos del universo de discurso al conjunto difuso. La elección de la forma de la función de pertenencia es subjetiva y dependiente del contexto. No obstante, por razones prácticas, se suelen emplear funciones triangulares, trapezoidales o en forma de campana como las que se muestran en la figura 1.9 [3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].

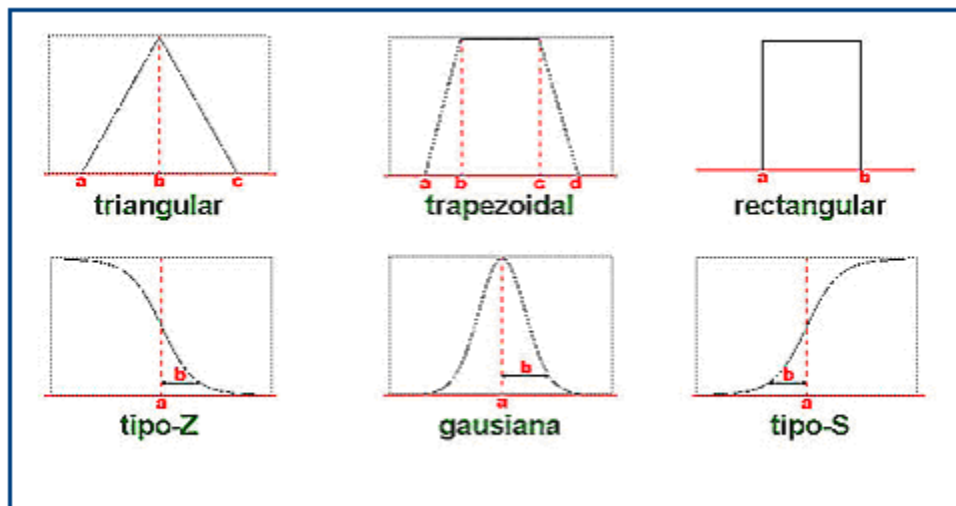


FIGURA 1.9. Ilustración de los tipos de funciones de pertenencia que definen los elementos de un conjunto difuso.

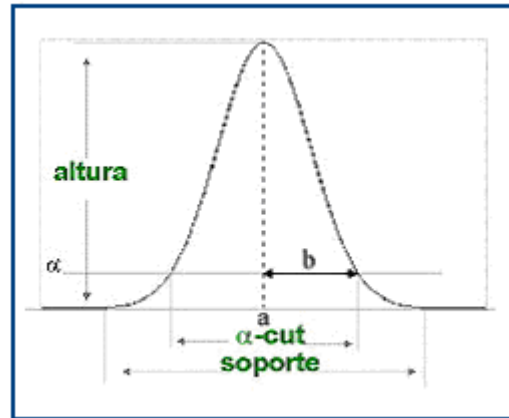


FIGURA 1.10. Parámetros necesarios para definir una función de pertenencia

1.8 FUZZY SINGLETON

Un conjunto difuso cuyo soporte contiene un único elemento, u_0 , es denominado una *singularidad difusa* o **fuzzy singleton**.

$$\mu_F(u) = 0 \text{ si } u \neq u_0 \quad \mu_F(u_0) = 1$$

Ec.2

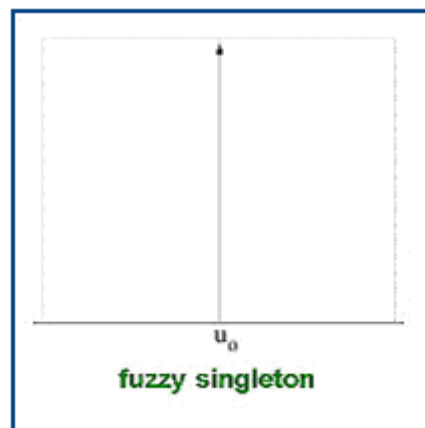


FIGURA 1.11. Ilustración de la singularidad difusa o Fuzzy singleton, en algunos textos llamado barra, el cual contiene un solo elemento como soporte.

El uso de *singletons* simplifica considerablemente el proceso de inferencia y posibilita la implementación electrónica eficiente de los sistemas de inferencia difusos [3.- FLEB, a Fuzzy logic E-Book, www.esime.edu.es].

1.9 OPERACIONES ENTRE CONJUNTOS DIFUSOS¹

La mayoría de las nociones básicas de la teoría clásica de conjuntos, como los conceptos de igualdad o inclusión, se pueden extender a los conjuntos difusos. Asimismo, la elección de los operadores “min” y “max” para representar la unión e intersección de conjuntos difusos está de acuerdo con la idea intuitiva de dichas operaciones. De esta forma, dados dos conjuntos difusos A y B definidos sobre el mismo universo de discurso U , las operaciones de unión, intersección y complemento pueden definirse, a partir de sus funciones de pertenencia, como:

1.9.1 UNION

La unión entre los conjuntos difusos A y B es un conjunto difuso cuya función de pertenencia para un elemento concreto del universo de discurso es la mayor de las funciones de pertenencia de A y B (operación “max”).

$$\mu_{A \cup B} = \max [\mu_A(u), \mu_B(u)], \forall u \in U$$

Ec. 3

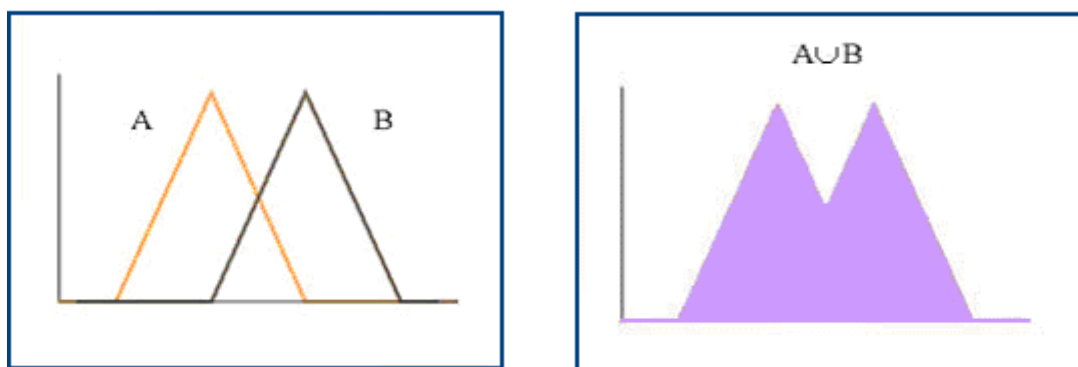


FIGURA 1.12. (Izquierda) Dados los conjuntos A y B , se presenta el resultado de $A \cup B$ (derecha) como el máximo valor entre A y B

¹ Toda la sección correspondiente a los conjuntos difusos y la operaciones entre ellos fue tomada de FLEB, a Fuzzy logic E-Book, www.esime.edu.es

1.9.2 INTERSECCIÓN

La intersección entre A y B es un conjunto difuso cuya función de pertenencia para un elemento concreto del universo de discurso es la menor de las funciones de pertenencia de A y B (operación “min”).

$$\mu_{A \cap B}(u) = \min[\mu_A(u), \mu_B(u)], \forall u \in U$$

Ec. 4

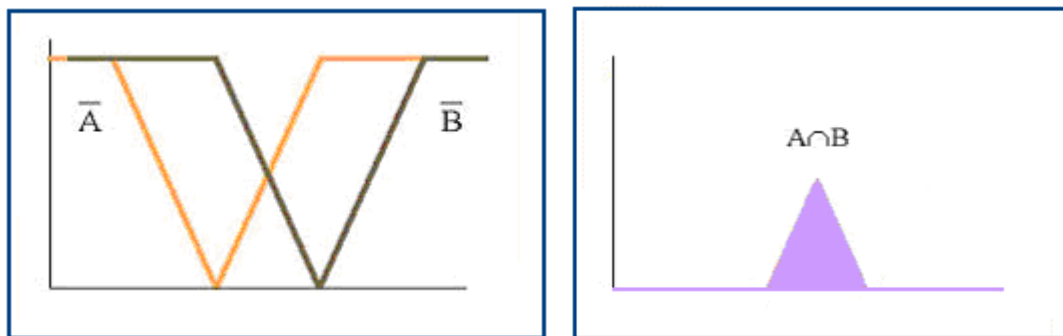


FIGURA 1.13. Dados los complementos de A y B (izquierda) la intersección $A \cap B$ (derecha) se define como el mínimo valor entre el complemento de A y el correspondiente de B

1.9.3 COMPLEMENTO

El complemento de un conjunto difuso A es otro conjunto difuso cuya función de pertenencia viene dada por: $\mu_{\bar{A}}(u) = 1 - \mu_A(u) \forall u \in U$

Ec. 5

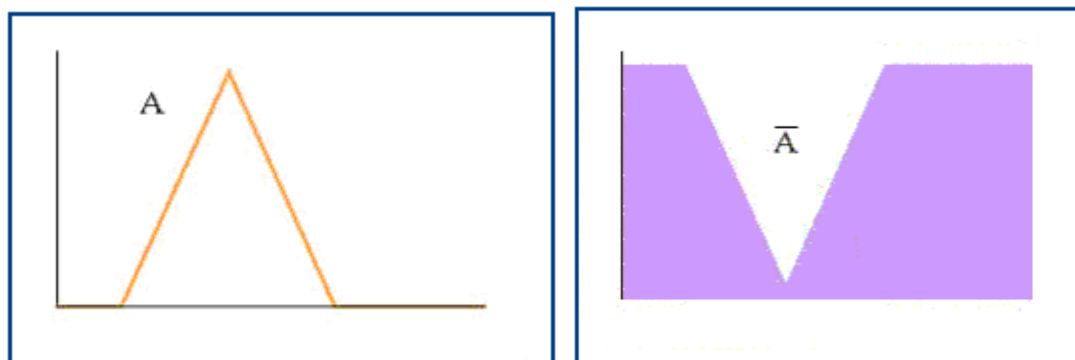


FIGURA 1.14. Dado un conjunto A , su complemento se define como la resta de cada valor puntual a 1.

Para el caso de las operaciones de unión e intersección de un conjunto difuso y su complemento es de tomar en cuenta las siguientes leyes:

1.9.3.1 LEY DEL TERCIO EXCLUSO

Esta ley enuncia que dado un conjunto difuso y su complemento, el resultado de la unión entre ambos es diferente al conjunto U (universo), y ello tiene su base en la operación “max” o unión de conjuntos difusos, en la cual dados dos conjuntos, la unión de estos es igual al máximo valor entre sus elementos, lo cual se ilustra en la figura 15 y matemáticamente se define por lo expresado en la ecuación 5.

$$A \cup \bar{A} \neq U$$

Ec. 5

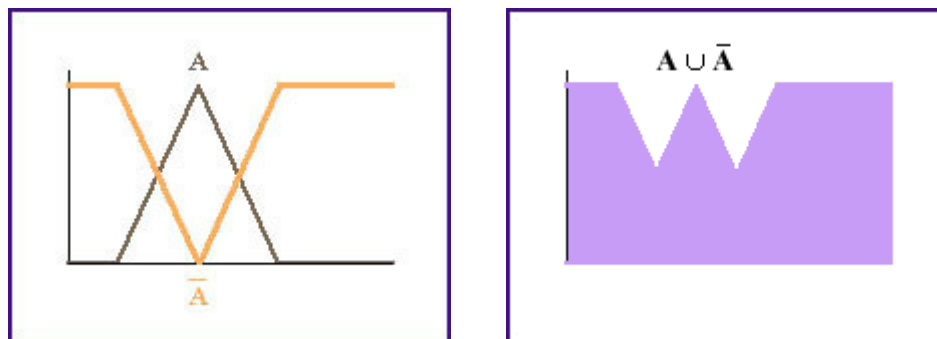


FIGURA 1.15. Dado el conjunto A y su complemento se ilustra la ley del tercio excluso donde la unión de estos es diferente al conjunto U (universo)

1.9.3.2 LEY DE LA CONTRADICCIÓN

En la teoría de conjuntos booleanos, la intersección de un conjunto con su complemento da como resultado un conjunto vacío, pero en el caso de un conjunto difuso y su complemento ello no se cumple, ya que si se hace referencia a la operación “min” que define la intersección entre dos conjuntos difusos, se comprenderá el porque el resultado

es diferente a un conjunto vacío, esto se ilustra gráficamente en la figura 1.16 y matemáticamente se define como:

$$A \cap \bar{A} \neq \emptyset$$

Ec. 6

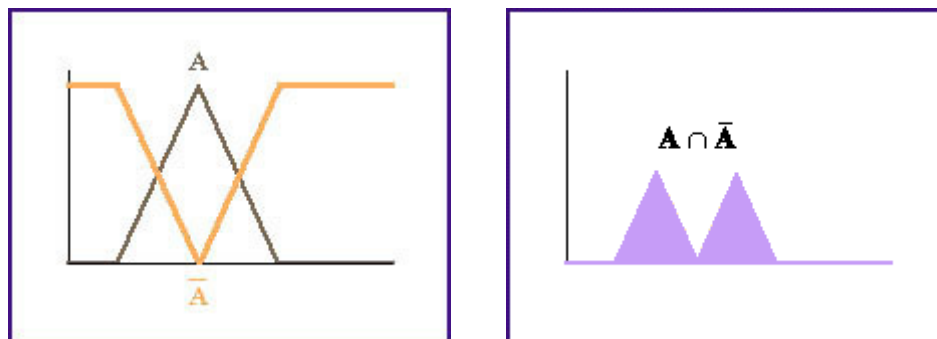


FIGURA 1.16. Dado un conjunto A y su complemento (Izquierda) el resultado de la intersección de estos (derecha)

1.9.4 INTERSECCION DIFUSA: T NORMAS

Una norma triangular o **T-norma** es una función $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ que cumple las siguientes propiedades:

Propiedad conmutativa	$T(x,y) = T(y,x), \forall x,y \in [0,1]$
Propiedad Asociativa	$T[x,(y,z)] = T[(x,y),z], \forall x,y,w,z \in [0,1]$
Monótona no decreciente	Si $x \leq y, w \leq z, \rightarrow T(x,w) \leq T(y,z), \forall x,y,w,z \in [0,1]$
Elemento neutro	$T(x,1) = x, \forall x \in [0,1]$
Elemento absorbente	$T(x,0) = 0, \forall x \in [0,1]$

Tabla 1.1. Conjunto de propiedades a las que responde la función difusa T-norma

Además de la función mínimo, las siguientes funciones cumplen también con las propiedades de T-norma

PRODUCTO ALGEBRAICO

$$T_p(x, y) = x \cdot y$$

Ec. 7

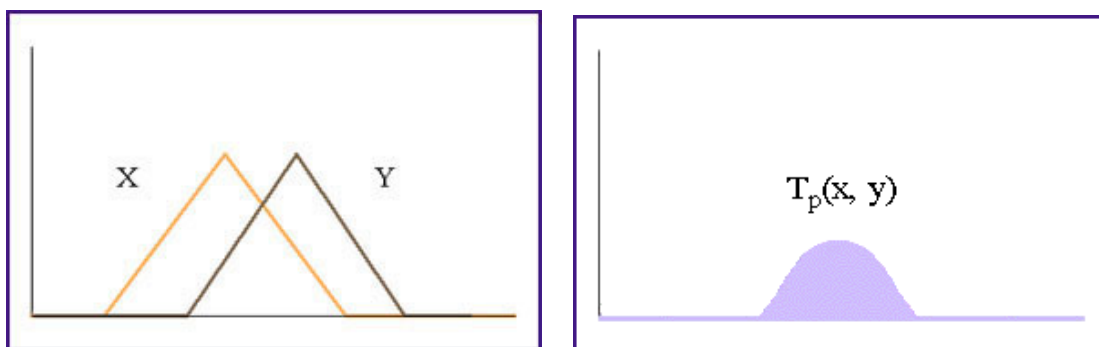


FIGURA 1.17. Dados los conjuntos X e Y (izquierda), el producto algebraico resultante (derecha)

PRODUCTO ACOTADO

$$T_{pa}(x, y) = \max \{0, x + y - 1\}$$

Ec. 8

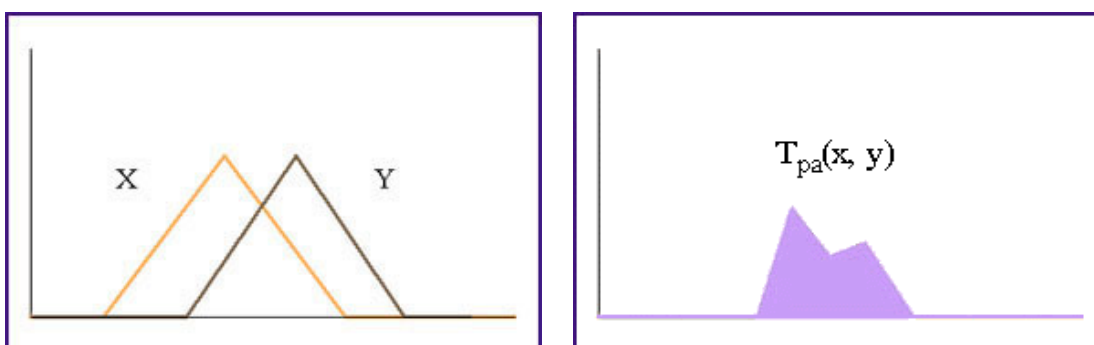


FIGURA 1.18. Dados los conjuntos difusos X e Y (izquierda) se ilustra gráficamente el producto acotado resultante (derecha)

PRODUCTO DRÁSTICO

$$T_{pd}(x, y) = \begin{cases} x, & \text{si } y = 1; \\ y, & \text{si } x = 1; \\ 0, & \text{si } x, y < 1 \end{cases}$$

Ec. 9

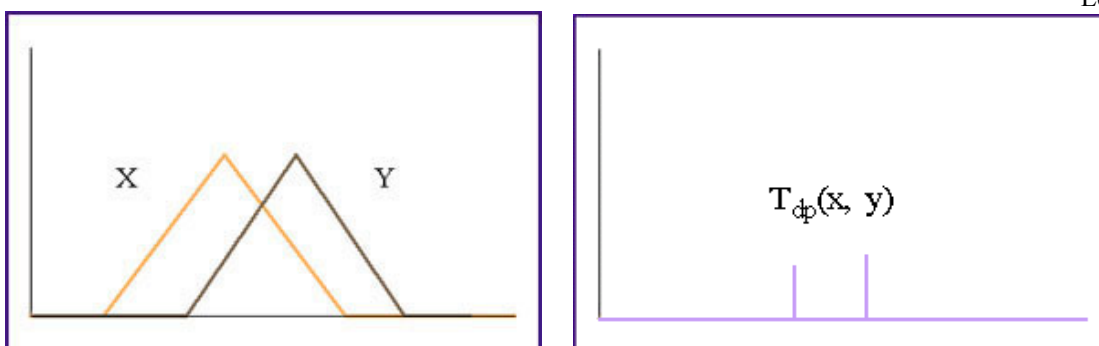


FIGURA 1.19. Producto drástico entre los conjuntos X e Y

1.9.5 UNION DIFUSA: S NORMAS

Una conorma triangular o **S-norma** es una función $S: [0, 1] \times [0, 1] \rightarrow [0, 1]$ que cumple las siguientes propiedades:

Propiedad conmutativa	$S(x,y) = S(y,x), \forall x,y \in [0,1]$
Propiedad Asociativa	$S[x,(y,z)] = S[(x,y),z], \forall x,y,w,z \in [0,1]$
Monótona no decreciente	Si $x \leq y, w \leq z, \rightarrow S(x,w) \leq S(y,z), \forall x,y,w,z \in [0,1]$
Elemento neutro	$S(x,1) = x, \forall x \in [0,1]$
Elemento absorbente	$S(x,0) = 0, \forall x \in [0,1]$

Tabla 1.2. Conjunto de propiedades a las que responde una función S-norma

Además de la función máximo, las siguientes funciones cumplen las propiedades S-norma

SUMA ALGEBRAICA

$$S_s(x, y) = x + y - xy$$

Ec. 10

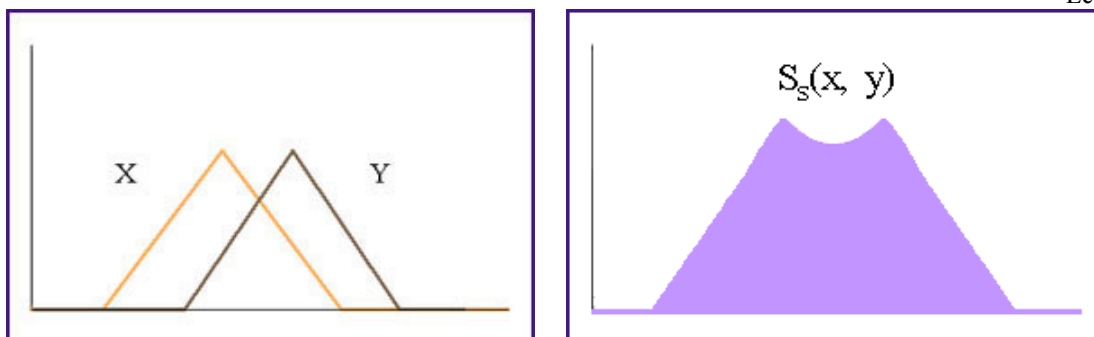


FIGURA 1.20. Ilustración de la suma algebraica entre dos conjuntos difusos X e Y.

SUMA ACOTADA

$$S_{sa}(x, y) = \min \{1, x + y\}$$

Ec. 11

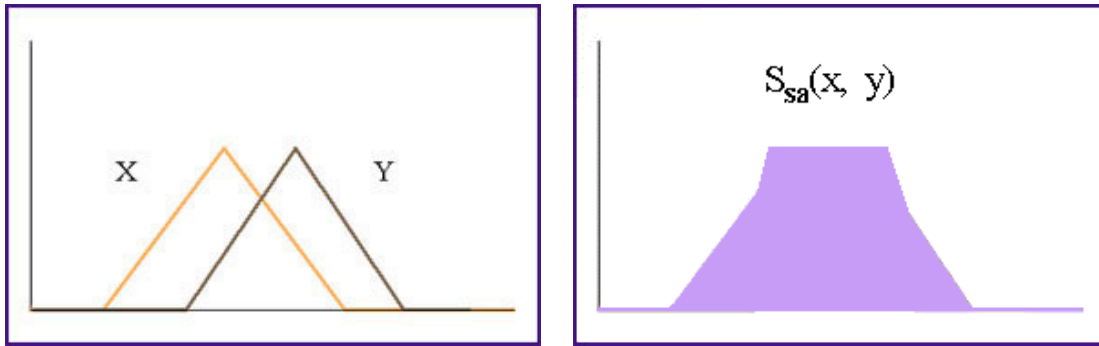


FIGURA 1.21. Dados los conjuntos difusos X e Y (izquierda) se ilustra el resultado de la suma acotada (derecha)

SUMA DRÁSTICA

$$S_{sd}(x, y) = x, \text{ si } y = 0;$$

$$y, \text{ si } x = 0;$$

Ec. 12

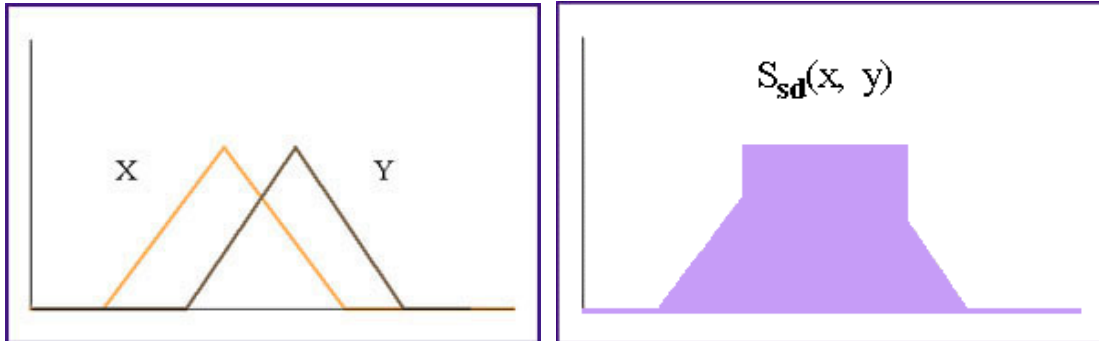


FIGURA 1.22. Representación gráfica de la suma drástica (derecha) entre X e Y

1.10 RELACIONES DIFUSAS

El concepto de **relación difusa** es una generalización del concepto de relación de la teoría clásica de conjuntos. Mientras que una relación entre dos conjuntos clásicos describe la existencia o no de asociación entre los elementos de ambos conjuntos, una relación difusa describe el grado de asociación o interacción entre los elementos de dos o más conjuntos difusos. En el caso discreto, la relación difusa puede representarse mediante una matriz, denominada **matriz relacional difusa**, cuyos elementos toman valores en el intervalo $[0, 1]$.

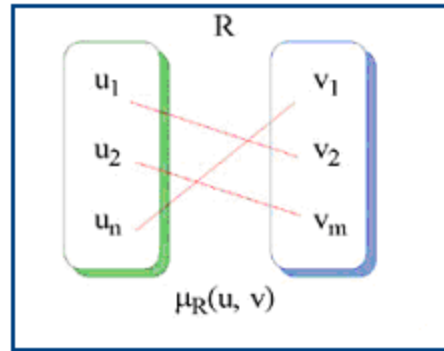


FIGURA 1.23. Dados los conjuntos u y v la relación $\mu_R(u, v)$ describe el grado de asociación entre los mismos.

1.11 COMPOSICIÓN DE RELACIONES DIFUSAS

Una relación difusa es un conjunto difuso en el espacio producto. Por tanto, es posible aplicar las operaciones algebraicas y de teoría de conjuntos usando los operadores unión, intersección y complemento definidos previamente, donde T y S pueden ser cualquiera de las T -normas y S -normas anteriores.

La composición de relaciones difusas en el mismo espacio producto permite interpretar sentencias de tipo “ x es mucho más grande que y y x es igual a y ”. Pero también es posible componer relaciones difusas definidas sobre espacios producto diferentes, como las que surgen al combinar las sentencias “ x es mucho más grande que y ” e “ y es igual a z ”. En este último caso la relación difusa resultante de la composición relacionará las variables x y z .

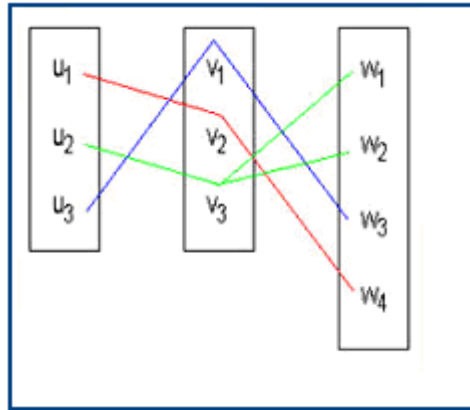


FIGURA 1.24. Dadas dos relaciones difusas R y S definidas en $U \times V$ y $V \times W$, respectivamente, la composición de R y S es una nueva relación difusa $Q = R \circ S$ definida en $U \times W$, de forma que (u, w) pertenecerá a Q si y sólo si existe al menos un elemento v perteneciente a V tal que (u, v) pertenezca a R y (v, w) pertenezca a S .

El concepto de composición de relaciones difusas definidas sobre espacios producto diferentes que comparten un universo de discurso se ilustra gráficamente en el diagrama de la figura 1.24.

1.12 REGLAS DIFUSAS²

Una **regla difusa** es una sentencia condicional del tipo IF-THEN, donde el *antecedente* y el *consecuente* son proposiciones difusas:

IF <proposición difusa> **THEN** <proposición difusa>

² Tomado de NILSSON Nils. J. Inteligencia Artificial Una Nueva Síntesis

1.12.1 PROPOSICIONES DIFUSAS

Existen dos tipos de proposiciones difusas

ATOMICAS

Asocian una etiqueta a una variable: *'la temperatura es alta'*.

COMPUESTAS

Se obtienen combinando proposiciones atómicas mediante los distintos **conectivos**

lingüísticos:	Conectivo “Y”	<i>El error es grande Y su variación pequeña</i>
	Conectivo “O”	<i>El error es grande O su variación pequeña</i>
	Conectivo “NO”	<i>El error NO es grande</i>

1.12.2 FUNCION DE IMPLICACIÓN

Un operador de implicación difusa o **función de implicación** expresa la relación que existe entre el antecedente y el consecuente de una regla. Una regla del tipo “*si x es A entonces y es B*” puede interpretarse como la relación difusa dada por:

$$\mu_{A \rightarrow B(x,y)} = \Phi [\mu_{A(x)}, \mu_{B(y)}] \quad \text{Ec.13}$$

Donde Φ es una función de implicación difusa.

Se han propuesto numerosas funciones de implicación difusas. La mayoría de ellas surgen como extensión de las implicaciones utilizadas en la lógica proposicional o en la lógica multivaluada (LMV) y de la elección de diferentes opciones para realizar las operaciones de intersección, unión y complemento de conjuntos.

1.12.3 FUNCIONES DE IMPLICACIÓN GLOBALES

En la lógica proposicional tradicional una proposición sólo puede tomar los valores verdadero “V” o falso “F”. Una expresión del tipo “*si p entonces q*” se representa mediante la implicación $(p \rightarrow q)$ que es equivalente a

$$(\neg p) \vee q \quad (p \wedge q) \vee (\neg p) \quad \text{Ec. 14}$$

Usando el isomorfismo existente entre la lógica proposicional y la teoría de conjuntos, los conectivos \vee , \wedge y \neg se pueden sustituir por los operadores \cup , \cap y $-$ (complemento). Puesto que en las reglas difusas p y q son proposiciones difusas, podemos interpretar la función de implicación difusa sustituyendo los operadores anteriores por S-normas, T-normas y C-normas, respectivamente. Formalmente, podemos establecer que, dadas dos variables lingüísticas x e y , definidas sobre los universos de discurso X e Y , y dos conjuntos difusos A y B en X e Y , la regla difusa “*si x es A entonces y es B*” es interpretada como una relación difusa en $X \times Y$, cuya función de pertenencia viene dada por:

$$\mu_{A \rightarrow B(x,y)} = S \{[\mu_A(x)], [\mu_B(y)]\} \quad \text{Ec. 15}$$

$$\mu_{A \rightarrow B(x,y)} = S \{T[\mu_A(x)], [\mu_B(y)], C[\mu_A(x)]\} \quad \text{Ec. 16}$$

FUNCION DE IMPLICACIÓN

La siguiente tabla muestra distintas funciones de implicación propuestas en la literatura relacionada con lógica difusa:

F. Implicación	Basada en	T	S	expresión
Dienes-Resher	$(\neg p) \vee q$	-	max	$\mu Rb(x,y) = \max [1-\mu A(x), \mu B(y),]$
Mizumoto	$(\neg p) \vee q$	-	+	$\mu RM(x,y) = 1-\mu A(x) + \mu A(x) \mu B(y),$
Lukasiewicz	$(\neg p) \vee q$		\oplus	$\mu Ra(x,y) = \min [1, 1-\mu A(x), \mu B(y),]$
Zadeh	$(p \wedge q) \vee (\neg p)$	min	max	$\mu RM(x,y) = \max \{ \min [\mu A(x), \mu B(y),], 1-\mu A(x) \}$
Goguen	LMV	-	-	$\mu R\Delta(x,y) = \{1, \text{si } \mu A(x) \leq \mu B(y); \mu A(x) / \mu B(y), \text{si } \mu A(x) > \mu B(y)\}$
Godel	LMV	-	-	$\mu Rg(x,y) = \{1, \text{si } \mu A(x) \leq \mu B(y); \mu B(y), \text{si } \mu A(x) > \mu B(y)\}$
Sharp	LMV	-	-	$\mu Rs(x,y) = \{1, \text{si } \mu A(x) \leq \mu B(y); 0, \text{si } \mu A(x) > \mu B(y)\}$
Mandami	$p \wedge q$	Min	-	$\mu Rc(x,y) = \min [\mu A(x), \mu B(y),]$
Larsen	$p \wedge q$	prod	-	$\mu p(x,y) = \mu A(x) \cdot \mu B(y),]$

Tabla 1.3 Resumen de las diferentes funciones de implicación

1.13 RAZONAMIENTO APROXIMADO³

Las técnicas de razonamiento permiten obtener deducciones lógicas a partir de proposiciones. El valor de verdad de una proposición difusa puede ser cualquier número en el intervalo $[0, 1]$. Esta generalización es la base de las **técnicas de razonamiento aproximado** que permiten deducir conclusiones correctas a partir de premisas vagas e imprecisas. Las reglas de inferencia usadas en lógica difusa son una generalización de las reglas de inferencia más importantes de la lógica tradicional. La figura 1.25 muestra la diferencia entre el **modus ponens** tradicional y el **modus ponens difuso o generalizado**. En este último caso A, B, A', B' son términos lingüísticos representados mediante conjuntos difusos y B' se aproximará tanto más a B cuanto más se aproxime A' a A .

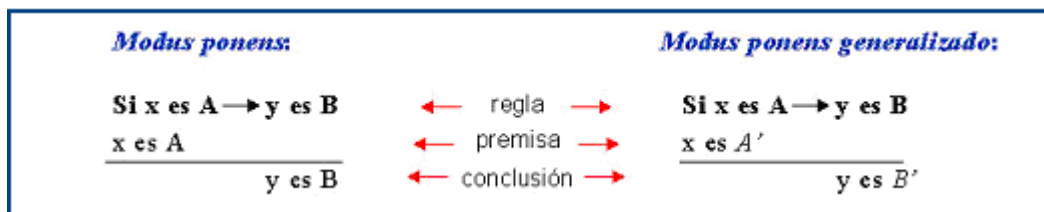


FIGURA 1.25. Modus ponens tradicional (izquierda), modus ponens generalizado

La **regla composicional de inferencia** propuesta por Zadeh permite evaluar numéricamente B' en función del grado de similitud entre A y A' :

³ tomado de FLEB, a Fuzzy logic E-Book. www.esime.edu.es

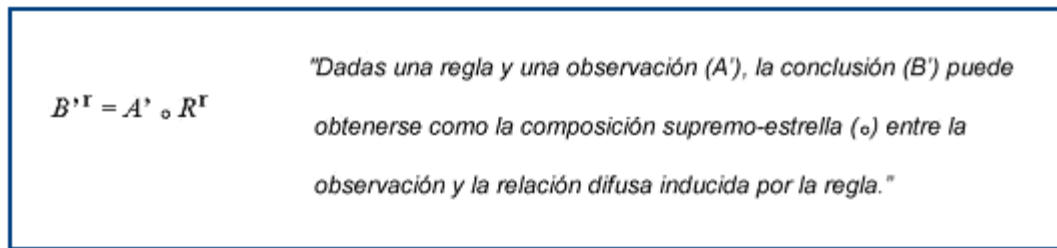


FIGURA 1.26. Regla composicional de inferencia formulada por Lofti A. Zadeh

La regla composicional de inferencia proporciona un mecanismo para evaluar el resultado de una regla difusa. Un sistema de inferencia difuso contendrá un conjunto de reglas de descripción lingüística. En el caso más general los antecedentes y consecuentes de estas reglas incluirán proposiciones difusas compuestas, es decir, combinarán múltiples entradas y salidas. Este tipo de sistema se denomina sistema MIMO (*"multiple input multiple output"*). No obstante, un sistema MIMO siempre puede ser considerado como un conjunto de sistemas con entradas múltiples y una única salida, o sistemas MISO (*"multiple input single output"*).

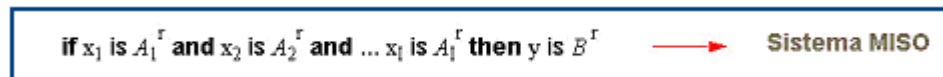


FIGURA 1.27. Ejemplo de una regla de inferencia de múltiples antecedentes unidos por el conectivo Y.

A diferencia de un sistema experto convencional, en un sistema difuso varias reglas pueden estar activas simultáneamente. La conclusión global se calculará por agregación de las soluciones parciales aportadas por cada regla. En este punto surgen dos preguntas:

¿Cómo interpretar el operador de agregación?

- Si cada una de las reglas es independiente de las demás, la agregación de reglas es la unión (R^1 o R^2 ... o R^n).

- Si las reglas están fuertemente acopladas (deben cumplirse las condiciones de todas las reglas para inferir un resultado) se utiliza la intersección (R^1 y R^2 ... y R^n).
- En aplicaciones de control se usa una S-norma para la agregación y funciones de implicación basadas en T-normas (Mamdani y Larsen). Para otras funciones de implicación (Godel o basadas en el cálculo proposicional) se emplea una T-norma para la agregación.

¿Qué procedimiento utilizar para realizar la agregación?

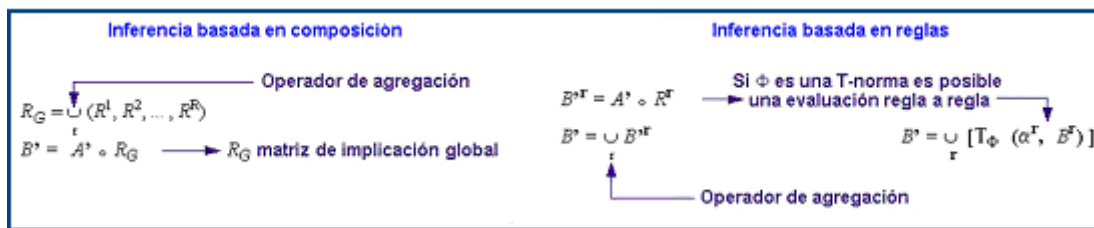


FIGURA 1.28. Ilustración de la inferencia que se debe utilizar para los casos de inferencia basada en composición (tablas de búsqueda) y la inferencia basada en reglas (si-entonces)

1.14 MECANISMOS DE INFERENCIA

1.14.1 MIN-MAX

El mecanismo de inferencia Min-Max (o **método de Mamdani**) es el que más veces aparece descrito en la literatura. Utiliza la T-norma mínimo como función de implicación y la S-norma máximo como operador de agregación. La función de pertenencia de la conclusión inferida al aplicar la base de reglas viene dada por la siguiente expresión (donde α^r es el grado de activación de la regla r-ésima):

$$\mu_{B^r}(y) = \max_r \{ \min (\alpha^r, \mu_{B^r}(y)) \}$$
Ec. 17

$$\alpha^r = \min_r (\mu_{A_1^r}(\bar{x}_1), \mu_{A_2^r}(\bar{x}_2) \dots \mu_{A_n^r}(\bar{x}_n))$$
Ec. 18

1.14.2 PRODUCTO SUMA

Propuesto por Mizumoto, este método utiliza los operadores producto y suma como función de implicación y operador de agregación, respectivamente. Siendo α^r el grado de activación de la regla r-ésima, la función de pertenencia del conjunto difuso resultante viene dada por las siguientes expresiones:

$$\mu_{B^r}(y) = \sum_r \alpha^r \cdot \mu_{B^r}(y)$$
Ec. 19

$$\alpha^r = \min_r (\mu_{A_1^r}(\bar{x}_1), \mu_{A_2^r}(\bar{x}_2) \dots \mu_{A_n^r}(\bar{x}_n))$$
Ec. 20

1.15 METODOS DE DEFUZZIFICACION⁴

En los mecanismos de inferencia, el resultado de la inferencia es un conjunto difuso. Para que esta información pueda utilizarse en determinadas aplicaciones, como las de control, es preciso obtener un valor concreto representativo de dicho conjunto.

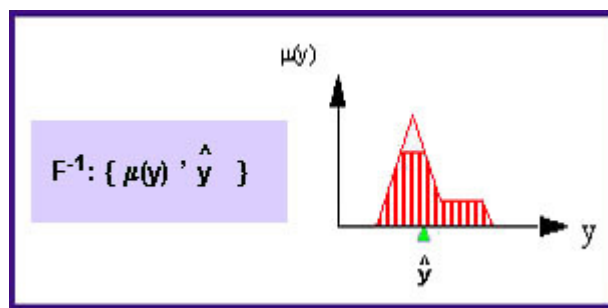


FIGURA 1.29. El proceso de **defuzzificación** se expresa mediante el operador defuzzificador F^{-1} que transforma la función de pertenencia representativa de un conjunto difuso $\mu(y)$ en un elemento concreto del universo de discurso \hat{y}

1.15.1 METODOS CONVENCIONALES

Para estos métodos la característica principal es que para obtener una salida defuzzificada, se hace necesario recorrer todo el universo de discurso de la variable de salida, estos métodos son:

- **Centro de gravedad (CoG)**

Determina el valor representativo de un conjunto difuso como el centro del área limitada por el conjunto difuso resultante de aplicar las diferentes reglas.

⁴ tomado de FLEB, a Fuzzy logic E-Book. www.esime.es

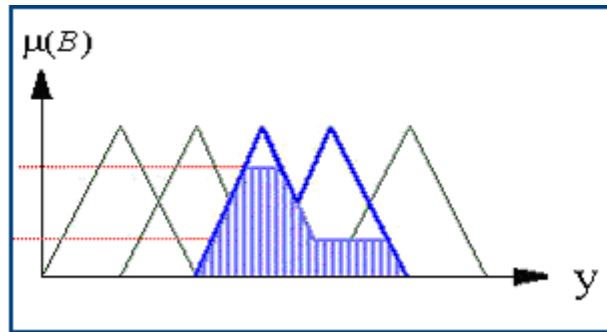


FIGURA 1.30. El método del centro de gravedad calcula el centroide del área sombreada y se asigna a \hat{y}

Matemáticamente el centro de gravedad se determina mediante la siguiente ecuación:

$$\text{CoG} \rightarrow \hat{y} = \frac{\sum_{i=1}^N y_i \cdot \mu_B(y_i)}{\sum_{i=1}^N \mu_B(y_i)}$$

Ec. 21

- **Centro de sumas (CoS)**

El objetivo de este método es contemplar la contribución de cada una de las conclusiones parciales. Para ello utiliza la suma en lugar del máximo como operador de agregación de reglas con lo que las áreas que se superponen se reflejan en el resultado, en la figura 33 se puede apreciar gráficamente el aporte que hace cada termino a la salida, matemáticamente el centro de sumas se determina por la ecuación 22.

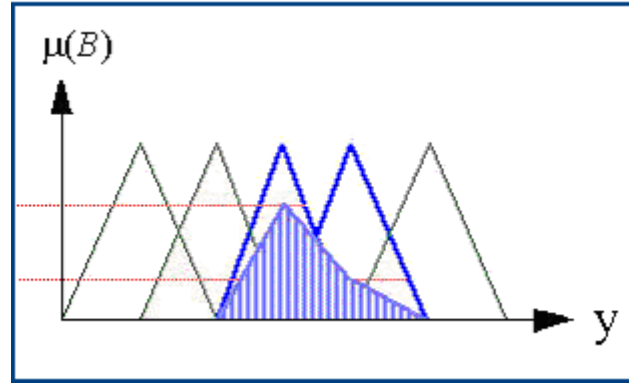


FIGURA 1.31. El método del centro de sumas determina el centroide del área sombreada que corresponderá a la salida defuzzificada \hat{y}

$$\text{CoS} \rightarrow \hat{y} = \frac{\sum_{i=1}^N y_i \cdot \sum_{r=1}^R \mu_B^r(y_i)}{\sum_{i=1}^N \sum_{r=1}^R \mu_B^r(y_i)} = \frac{\sum_{r=1}^R \alpha^r \cdot \sum_{i=1}^N y_i \cdot \mu_B^r(y_i)}{\sum_{r=1}^R \alpha^r \cdot \sum_{i=1}^N \mu_B^r(y_i)} \quad \text{Ec. 22}$$

- **Primer Máximo (FoM)**

Utilizando el conjunto difuso obtenido tras la agregación de todas las reglas, este método toma como resultado el valor más pequeño del dominio de salida que presenta grado de pertenencia máximo, y matemáticamente se describe por la ecuación 23.

$$\text{FoM} \rightarrow \hat{y} = \inf \{y_j \in Y \mid \mu_B(y_j) = \max_i \mu_B(y_i)\} \quad \text{Ec. 23}$$

- **Ultimo Máximo (LoM)**

Utilizando el conjunto difuso obtenido tras la agregación de todas las reglas, este método toma como resultado el valor más grande del dominio de salida que presenta grado de pertenencia máximo y se rige mediante la siguiente ecuación:

$$\text{LoM} \rightarrow \hat{y} = \sup \{y_j \in Y \mid \mu_B(y_j) = \max_i \mu_B(y_i)\}$$

Ec. 24

1.15.2 METODOS SIMPLIFICADOS

Sustituyen la información difusa de los consecuentes de las reglas por una serie de parámetros.

1. Media Difusa (FM)

Consiste en elegir como conclusión parcial de cada regla el punto de máxima pertenencia.

La salida se obtiene como la suma de las conclusiones parciales de las diferentes reglas c^r ponderada por los grados de activación α^r como puede verse en la figura 1.32.

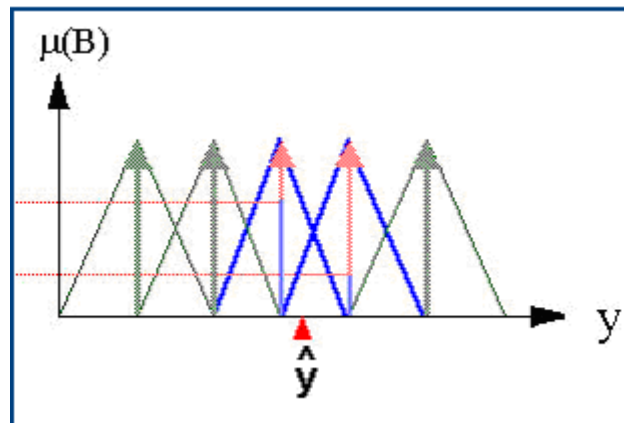


FIGURA 1.32. Representación gráfica de una salida con elementos tipo singleton, donde a cada uno le corresponde cierto grado de verdad como consecuencia de sus respectivos antecedentes.

$$\hat{y} = \frac{\sum_{r=1}^R \alpha^r \cdot c^r}{\sum_{r=1}^R \alpha^r}$$

Ec. 25

2. Media Difusa Ponderada (WFM)

Este método, al igual que CoS y QM, obtiene la salida defuzzificada mediante la suma de las conclusiones parciales de cada regla (c^r) ponderada por el grado de activación (α^r) y por un segundo parámetro (γ_r) el cual se define para cada método como sigue en la ecuación 26.

$$\hat{y} = \frac{\sum_{r=1}^R \alpha^r \cdot c^r \cdot \gamma_r}{\sum_{r=1}^R \alpha^r \cdot \gamma_r}$$

$\gamma = \text{área}$	➡	Centro de Sumas (CoS)
$\gamma = \text{soporte}$	➡	Media Difusa Ponderada (WFM)
$\gamma = 1/\text{soporte}$	➡	Método de Calidad (QM)

Ec. 26

3. Método de calidad (QM)

Este método, al igual que WFM y QM, obtiene la salida defuzzificada mediante la suma de las conclusiones parciales de cada regla (c^r) ponderada por el grado de activación (α^r) y por un segundo parámetro (γ_r), este método al igual que la media difusa pondera se gobierna por la ecuación 26.

4. Método “level grading” (LGM)

El método LGM trata de evitar el problema de dominancia de las funciones de pertenencia anchas utilizando una medida de certidumbre del α -cut inversamente proporcional al soporte del α -cut (la anchura de la función de pertenencia para un grado de activación determinado).

Para el caso particular en que las funciones de pertenencia sean triangulares, la relación entre la base del triángulo (d) y su anchura para un grado de activación $\alpha(d')$ es $d' = d \cdot (1 - \alpha)$ y la salida proporcionada por este método viene dada por la expresión siguiente:

$$\text{LGM} \rightarrow \hat{y} = \frac{\sum_{r=1}^R \alpha^r \cdot c^r / d_r \cdot (1 - \alpha)}{\sum_{r=1}^R \alpha^r / d_r \cdot (1 - \alpha)}$$

Ec. 27

5. Método de Yager (YM)

Se diferencia de los demás métodos en que calcula dinámicamente los valores de c^r . Para ello utiliza la media aritmética de las proyecciones sobre el universo de salida de los puntos de corte del grado de activación de una regla y la función de pertenencia de su consecuente.

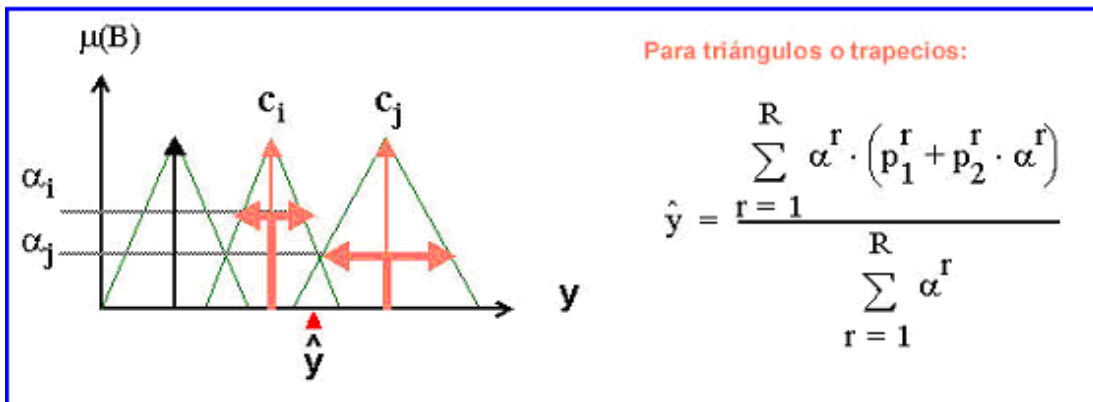


FIGURA 1.33 y Ecuación 28. Ilustración del método Yager para defuzzificación de las salidas (derecha) y la ecuación que lo define.

1.16 ESTRUCTURA DE UN SISTEMA DIFUSO

Desde el punto de vista de implementación un sistema difuso incluye los bloques que aparecen ilustrados en la figura 1.34. El núcleo del sistema está formado por una **base de conocimiento** que contiene la definición de las funciones de pertenencia de antecedentes y consecuentes utilizadas en las reglas, y un **motor de inferencia** capaz de procesar dicha información de acuerdo con alguno de los mecanismos descritos en los apartados anteriores. Además de estos elementos básicos, en la mayoría de los casos es necesario incluir dos bloques de interfaz que conectan el motor de inferencia con las entradas y salidas del sistema. Estos bloques se denominan **fuzzificador** y **defuzzificador**, respectivamente.

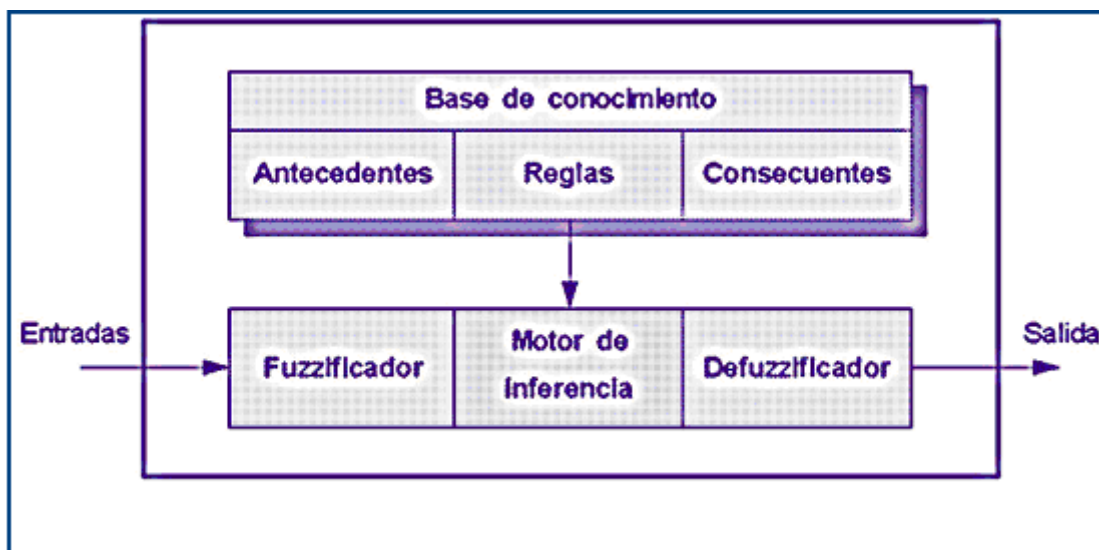


FIGURA 1.34. Detalle de los componentes de un Controlador Difuso, se muestra la interrelación entre la parte física (hardware y software) con la cognoscitiva

1.17 APLICACIONES INDUSTRIALES⁵

El éxito de las aplicaciones de control difuso ha sido el resultado de combinar, por un lado, los conceptos y estructuras generales de la teoría de control convencional, por otro, las técnicas de inferencia y representación del conocimiento mediante reglas del campo de la inteligencia artificial y, por último, el marco teórico que ofrece la lógica difusa para representar el conocimiento expresado con variables lingüísticas. A continuación se resumen las diferencias y analogías entre los distintos tipos de controladores convencionales y difusos:

1.17.1 CONTROL CONVENCIONAL

Las estrategias de control convencional que más extensamente se han aplicado se basan en el control PID (proporcional, integral y derivativo) y los tipos derivados de éste (P, PD, PI).

En el caso del **control proporcional (P)**, la acción de control es directamente proporcional al error instantáneo:

$$U(t) = K_p \cdot e(t) \quad \text{Ec. 29}$$

En el método de **control proporcional y derivativo (PD)**, la acción de control incorpora un término proporcional a la derivada del error:

$$U(t) = K_p \cdot e(t) + K_d \cdot de(t)/dt \quad \text{Ec. 30}$$

⁵ tomado de NILSSON Nils. J Inteligencia Artificial Una Nueva Síntesis

En el **control proporcional, integral y derivativo (PID)**, la acción de control incorpora un tercer término para integrar el error a lo largo de un intervalo de tiempo:

$$U(t) = K_p.e(t) + K_i.\int e(t)dt + K_d.de(t)/dt \quad \text{Ec. 31}$$

1.17.2 CONTROL DIFUSO

Los controladores difusos heredan de la teoría de control convencional la idea de incluir elementos proporcionales, integrales y derivativos.

La base de reglas de un **controlador Fuzzy-P** define la acción de control en función del error observado:

$$U(nT) = F(e(nT)) \quad \text{Ec. 32}$$

En un **controlador Fuzzy-PD** los antecedentes de las reglas combinan los valores del error y la variación del error:

$$U(nT) = F(e(nT)\Sigma e(nT)) \quad \text{Ec. 33}$$

1.18 SOPORTE PARA LOGICA DIFUSA CON EL MCU 68HC12⁶

La CPU12 tiene el primer conjunto de instrucciones de un microcontrolador para dirigirse específicamente a las necesidades de la lógica difusa.

La CPU12 incluye cuatro instrucciones que realizan tareas específicas de lógica difusa. Además, varias otras instrucciones son especialmente útiles en programas de lógica difusa. La amigabilidad global tipo C del conjunto de instrucciones también ayuda al desarrollo de programas eficientes de lógica difusa.

Las cuatro instrucciones de lógica difusa son

- MEM, (determina el grado de pertenencia), la cual evalúa funciones de pertenencia trapezoidales;
- REV (evaluación de reglas de lógica difusa) y REVW (evaluación ponderada de reglas de lógica difusa), las que realizan evaluación MIN-MAX de reglas no ponderada o ponderada; y
- WAV, la que realiza defuzzificación por promedio ponderado con funciones de pertenencia de salida tipo singleton (barra).

Otras instrucciones que son útiles para programas de lógica difusa del cliente incluyen:

- MINA (coloca el menor de dos valores sin signo de 8 bits en el acumulador A),
- EMIND (coloca el menor de dos valores sin signo de 16 bits en el acumulador D),
- MAXM (coloca el mayor de dos valores sin signo de 8 bits en memoria),
- EMAXM (coloca el mayor de dos valores sin signo de 16 bits en memoria),
- TBL (Búsqueda en tablas e interpolación),
- ETBL (Búsqueda extendida en tablas e interpolación), y

⁶ tomado de MC68HC12 Advanced Information, Motorola Semiconductors

· EMACS (Multiplicación y Acumulación extendida, con signo; 16*16 bits para obtener 32 bits).

Las adiciones a la lógica actual requeridas para implementar el soporte a la lógica difusa en la CPU12 son muy pequeñas, de modo que no hay incremento apreciable en costo para el usuario típico. Un núcleo de inferencia fuzzy para la CPU12 requiere un quinto a lo más de espacio de código, y se ejecuta quince veces más rápido que un núcleo comparable implementado en un microcontrolador típico de rango medio.

Al incorporar el soporte de lógica difusa a una familia de productos de microcontroladores de propósito general, Motorola ha hecho que la lógica difusa esté disponible para una enorme base de aplicaciones.

1.18.1 Fuzzyficación (MEM)

Durante el paso de fuzzyficación, los valores de entrada actuales del sistema son comparados contra funciones de pertenencia de entrada almacenadas para determinar el grado que cada etiqueta de entrada del sistema es verdad. Esto se logra hallando el valor y para el valor actual de entrada en una función de pertenencia trapezoidal para cada etiqueta de cada entrada del sistema.

La instrucción MEM en la CPU12 realiza este cálculo para una etiqueta de una entrada del sistema. Para realizar la tarea de fuzzyficación completa para un sistema, varias instrucciones MEM deben ser ejecutadas, usualmente en una estructura de lazo del programa.

La figura 1.35 exhibe un sistema de tres funciones de pertenencia de entrada, una para cada etiqueta de la entrada del sistema. El eje x de las tres funciones de pertenencia

representa el rango de valores posibles de la entrada del sistema. La línea vertical a través de las tres funciones de pertenencia representa a un valor específico de entrada del sistema. El eje y representa el grado de verdad y varia desde completamente falso (\$00 o 0%) hasta completamente verdad (\$FF o 100%).

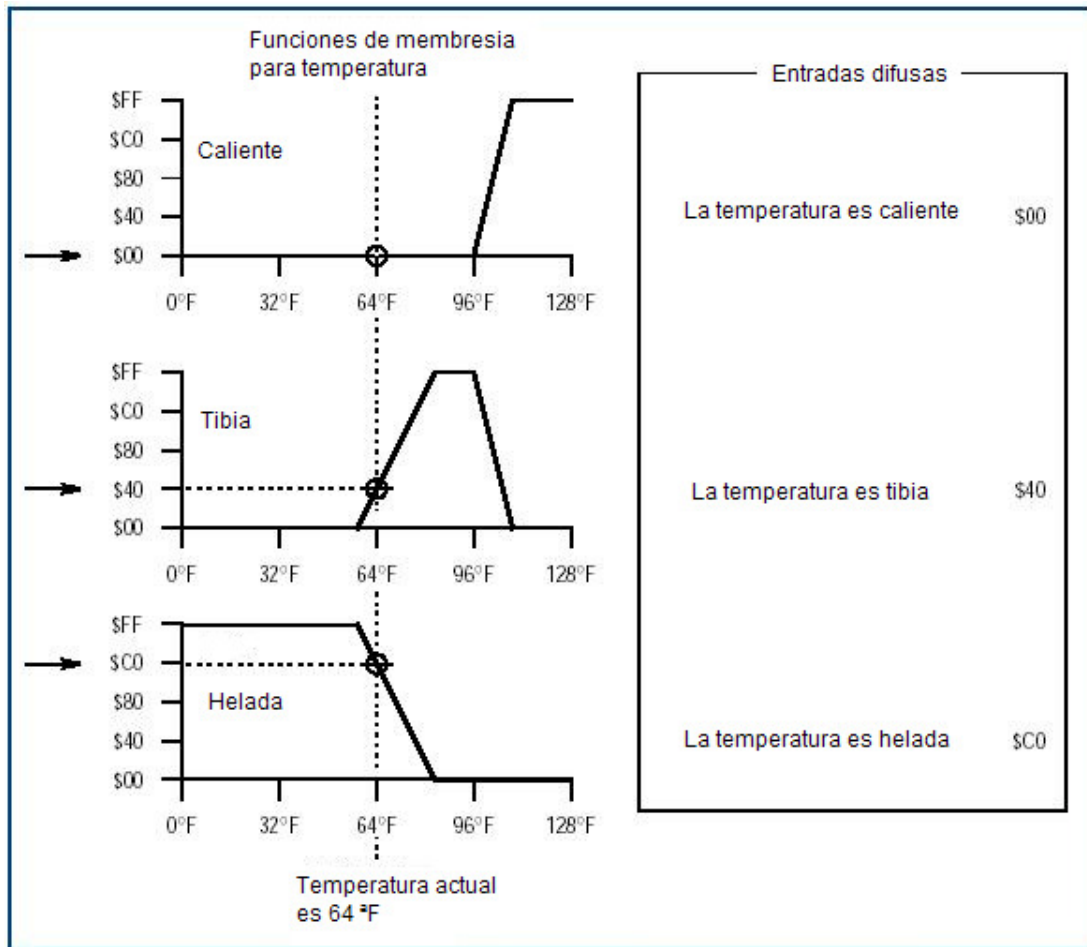


FIGURA 1.35. Funciones de membresía para la variable humedad implementadas en el MCU 68HC12, con un universo de discurso de 0°F a 128°F

El valor y donde la línea vertical intercepta a cada una de las funciones de pertenencia, es el grado al cual el valor actual de entrada concuerda con la etiqueta asociada para esta entrada del sistema. Por ejemplo, la expresión “la temperatura es tibia” es 25% verdad (\$40). El valor \$40 es almacenado en una localidad RAM, y es llamado una entrada

difusa (en este caso, la entrada difusa para “la temperatura es tibia”). Hay una localidad RAM para cada entrada difusa (por cada etiqueta de cada entrada del sistema).

Cuando el paso de fuzzyficación empieza, el valor actual de la entrada del sistema está en un acumulador de la CPU12, un registro índice apunta a la primera definición de función de pertenencia en la base de conocimiento, y un segundo registro índice apunta a la primera entrada difusa en RAM. En cuanto cada entrada difusa es calculada al ejecutar una instrucción MEM, el resultado es almacenado en la entrada difusa y ambos apuntadores son actualizados automáticamente para apuntar a las localidades asociadas con la siguiente entrada difusa. La instrucción MEM se encarga de todo excepto de contar el número de etiquetas por entrada del sistema y de cargar el valor actual de cualquiera entrada subsecuente del sistema.

El resultado final del paso de fuzzyficación es una tabla de entradas difusas representando las condiciones actuales del sistema.

1.18.2 Evaluación de Reglas (REV y REVW)

La evaluación de reglas es el elemento central de un programa de inferencia de lógica difusa. Este paso procesa una lista de reglas de la base de conocimiento usando valores actuales de entradas difusas en RAM para producir una lista de salidas difusas en RAM. Dichas salidas difusas pueden considerarse como sugerencias algo burdas de lo que la salida del sistema debería ser en respuesta a las actuales condiciones de entrada. Antes que los resultados puedan aplicarse, las salidas difusas deben de procesarse posteriormente, o ser defuzzyficadas, para producir un solo valor de salida que representa el efecto combinado de todas las salidas difusas.

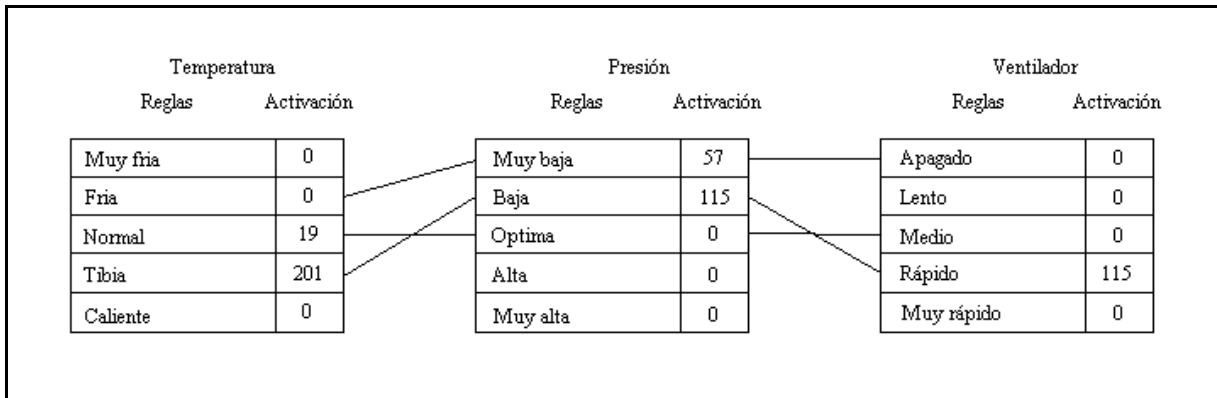


Figura 1.36. Ejemplo de la función REV para evaluar los antecedentes sobre la base de las reglas definidas para el controlador difuso.

La CPU12 ofrece dos variaciones de instrucciones para evaluación de reglas. La instrucción **REV** provista para reglas no ponderada (todas las reglas se consideran de igual importancia) según se muestra en la figura 1.36, donde puede verse la evaluación de los siguientes postulados:

“Si la temperatura es fría y la presión es muy baja entonces el ventilador debe estar apagado”.

“Si la temperatura es normal y la presión óptima entonces el ventilador debe estar a medio”.

“Si la temperatura es tibia y la presión es baja entonces el ventilador debe estar rápido”.

La instrucción **REVW**, similar, pero permite a cada regla tener un factor de peso separado el cual es almacenado en una estructura separada paralela de datos en la base de conocimiento. Aparte de los pesos, las dos instrucciones para evaluación de reglas también difieren en la manera que las reglas son codificadas en la base de conocimiento, según se observa en la figura 1.37 donde están definidos los mismos postulados

anteriores pero ahora incluyendo el nivel de importancia asignado a cada regla el cual afectará el valor de la salida defuzificada.

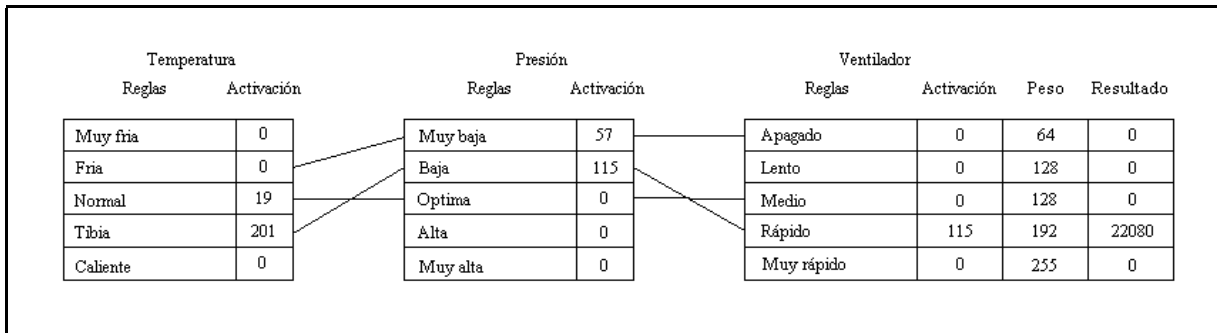


Figura 1.37. Ejemplo de la función REVW usada para evaluar los antecedentes de las reglas del controlador difuso.

A primera vista, parece que codificar esta regla en una forma compacta comprensible para el microcontrolador sería difícil, pero es realmente simple reducir la regla a una lista breve de punteros hacia localidades de memoria RAM como se muestra en el siguiente ejemplo:

```

;Define los términos de entrada para la variable Setpoint
spcero:      equ $0
spmuylento:  equ $1
splento:     equ $2
spmedio:     equ $3
sprapido:    equ $4
;Define los términos de entrada para la variable Floop
flcero:      equ $5
flmuylento:  equ $6
fllento:     equ $7
flmedio:     equ $8
flrapido:    equ $9
;Define los términos de salidas tipo singletons
cero:        equ $A
muy poco:    equ $B
poco:        equ $C
medio:       equ $D
todo:        equ $E

```

Donde cada una de las etiquetas de los términos tanto de entrada como de salida están definidos como una constante que luego sumada al valor del registro Y, permitirán obtener la dirección efectiva de la localidad de memoria deseada.

La porción izquierda de la regla es un postulado de condiciones de la entrada y la porción derecha de la regla es un postulado de acciones de salida. La porción izquierda de una regla está formada de uno o más (en este caso dos) antecedentes conectados por un operador fuzzy **AND**. Cada expresión antecedente consiste del nombre de la entrada del sistema, seguido por **es**, seguido por un nombre de etiqueta como puede verse en el siguiente postulado de ejemplo:

“SI setpoint medio Y floop rápido salida ES cero”

Cada etiqueta debe estar definida mediante una función de pertenencia en la base de conocimiento, y, cada expresión antecedente debe corresponder a una de las entradas difusas en RAM. Puesto que **Y** es el único operador permitido para conectar expresiones antecedentes, no hay necesidad de incluir a éstos en la codificación de la regla. Los antecedentes pueden codificarse como una simple lista de apuntadores hacia (o direcciones de) las entradas difusas a las cuales ellos hacen referencia como puede verse en los siguientes ejemplo donde cada una de las etiquetas se han definido previamente:

```
org $0930 ;define las reglas de inferencia difusas
dc.b spcero, flcero, sep, cero, sep
dc.b spmuylento, flcero, sep, muypoco, sep
dc.b splento, flcero, sep, poco, sep
dc.b spmedio, flcero, sep, medio, sep
dc.b sprapido, flcero, sep, todo, sep
dc.b end_rules
```

Lo que generaría el siguiente código para el MCU68HC12, en base a la tabla de igualdades mostrada previamente que define las etiquetas como valores constantes.

```
=00000930
0930 00 05 FE 0A FE
0949 01 05 FE 0B FE
0962 02 05 FE 0C FE
097B 03 05 FE 0D FE
0994 04 05 FE 0E FE
09AC FF
```

La porción derecha de una regla se forma de uno o más (en este caso uno) consecuentes. Cada expresión de consecuencia consiste del nombre de una salida del sistema, seguido por “es”, seguido por un nombre de etiqueta. Cada expresión de consecuencia corresponde a una salida difusa específica en RAM. Consecuencias para una regla pueden codificarse como una simple lista de apuntadores a (o direcciones de) las salidas difusas a las cuales hacen referencia.

Las reglas completas son almacenadas en la base de conocimiento como una lista de apuntadores o direcciones de entradas difusas y salidas difusas. Para que trabaje la lógica de evaluación de reglas, debe haber algunos medios de saber cuales apuntadores se refieren a entradas difusas, y cuáles se refieren a salidas difusas.

También debe haber una manera de saber cuando la última regla del sistema ha sido alcanzada.

Un método de organización es tener un número fijo de reglas con un número específico de antecedentes y consecuencias. Un segundo método, empleado en núcleos de Motorola Freeware M68HC11, es marcar el fin de la lista de reglas con un valor reservado, y usar un bit en los apuntadores para distinguir antecedentes de consecuentes. Un tercer método de organización, usado en la CPU12, es marcar el fin de la lista de reglas con un valor reservado, y separar antecedentes y consecuencias con otro valor reservado. Ello permite cualquier número de reglas, y permite a cada regla tener cualquier número de antecedentes y consecuencias, sujeto a los límites impuestos por la disponibilidad de memoria en el sistema.

Cada regla es evaluada secuencialmente, pero las reglas como grupo son tratadas como si ellas fueran todas evaluadas simultáneamente. Dos operaciones matemáticas toman lugar

durante la evaluación de reglas. El operador fuzzy **AND** corresponde a la operación matemática de hallar el mínimo y la operación fuzzy **OR** corresponde a la operación matemática de hallar el máximo. La operación fuzzy **AND** es usada para conectar antecedentes dentro de una regla. La operación fuzzy **OR** está implícita entre reglas sucesivas. Antes de evaluar cualquier Regla, todas las salidas difusas se ponen a cero (lo que significa nada de verdad). Cuando cada regla es evaluada, el más pequeño (el mínimo) antecedente es tomado para ser la verdad global de la regla. Este valor de verdad de la regla es aplicado a cada consecuencia de la regla (al almacenar este valor a la correspondiente salida difusa) a menos que la salida difusa es ya más grande (máxima). Si dos o más reglas afectan a la misma salida difusa, la regla que es más verdadera gobierna el valor en la salida difusa porque las reglas se conectan por un fuzzy **OR** implícito. Si se hace referencia al ejemplo de la figura 1.38 que muestra gráficamente los postulados:

“SI temperatura normal Y presión muy baja ventilador ES medio”

“SI temperatura tibia Y presión baja ventilador ES medio”

Dicha situación se resuelve aplicando la función AND difusa o **MIN** donde se toma como el valor de verdad el mínimo entre “temperatura normal” y “presión muy baja” para el primer postulado y “temperatura tibia” y “presión baja”, lo cual producirá los valores de 19 y 115 respectivamente; Pero, según puede observarse ambos casos producen como consecuente “ventilador medio”, ello se resuelve encontrando aplicando una función OR difusa o **MAX**, que toma como valor de verdad el máximo entre 19 y 115.

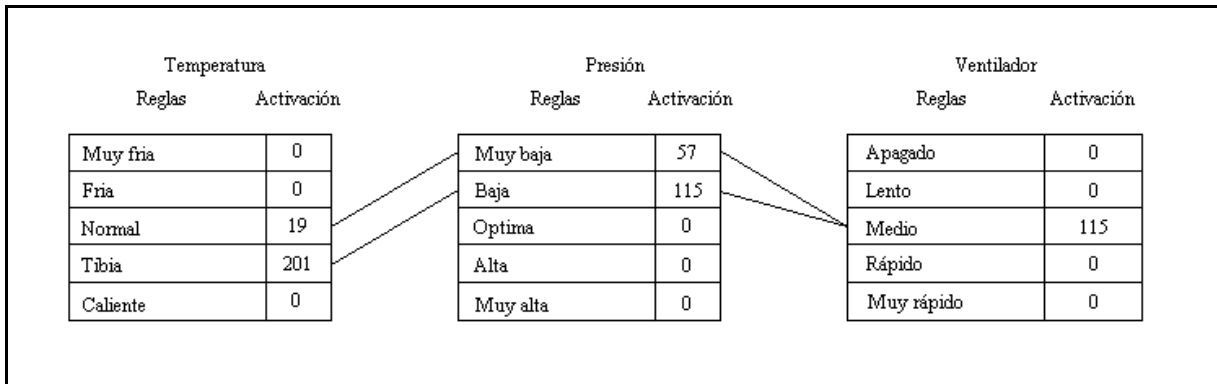


Figura 1.38. Ilustración de la forma como el MCU68HC12 resuelve la conexión entre antecedentes y consecuentes, así como entre reglas.

En el caso de reglas ponderadas, el valor de verdad para una regla se determina del modo usual hallando el más pequeño antecedente de regla. Antes de aplicar este valor de verdad a las consecuencias para la regla, el valor se multiplica por una fracción de cero (regla no habilitada) a uno (regla totalmente habilitada). El resultado modificado del valor de verdad se aplica entonces a las salidas difusas.

El resultado final del paso de evaluación reglas es una tabla de salidas difusas sugeridas o “crudas” en RAM. Estos valores fueron obtenidos al alimentar condiciones actuales (valores de entradas difusas) en las reglas del sistema en la base de conocimiento. Estos resultados crudos no pueden ser suministrados directamente a las salidas del sistema porque ellos pueden ser ambiguos. Por ejemplo, una salida cruda puede indicar que la salida del sistema debería ser media con a grado de verdad de 50% mientras, al mismo tiempo, otra indica que la salida del sistema debería ser baja con a grado de verdad de 25%. El paso de defuzzyficación resuelve tales ambigüedades.

1.18.3 Defuzzyficación (WAV)

El paso final en el programa de lógica difusa combina las salidas difusas crudas en una salida compuesta del sistema. A diferencia de las formas trapecoide usadas para las entradas, la CPU12 típicamente usa barras (singletons) para funciones de pertenencia de salida. Al igual que con las entradas, el eje x representa el rango de posibles valores para la salida del sistema. Las funciones de pertenencia tipo barra (Singleton) consisten de las posiciones en el eje x para cada etiqueta de la salida del sistema. Las salidas difusas corresponden a la altura en el eje y de la correspondiente función de pertenencia de salida.

La instrucción WAV calcula las sumas del numerador y del denominador para el promedio ponderado de la salida difusa de acuerdo con la fórmula:

$$salida\ del\ sistema = \frac{\sum_{j=1}^n S_j F_j}{\sum_{j=1}^n F_j}$$

Ec. 34

Donde n es el número de etiquetas de la salida del sistema, S_j son las posiciones de los singleton de la base de conocimiento, y F_j son las salidas difusas en RAM. Para un programa común de lógica difusa en la CPU12, n es ocho o menos (aunque esta instrucción puede manejar cualquier valor hasta 255) y S_j y F_j son valores de 8 bits. La división final es realizada con una instrucción separada EDIV colocada inmediatamente después de la instrucción WAV.

Antes de ejecutar WAV, un acumulador debe ser cargado con el número de iteraciones (n), un registro índice debe apuntar a la lista de posiciones singleton en la base de conocimiento, y un segundo registro índice debe apuntar a la lista de salidas difusas en RAM. Si el sistema tiene más que una salida del sistema, la instrucción WAV es ejecutada una vez por cada salida del sistema.

2. INTERFAZ GRAFICA

La interfaz gráfica ha sido desarrollada en Visual Basic, el cual está definido como un lenguaje de programación de cuarta generación (4GL) el cual permite un Desarrollo Rápido de Aplicaciones. En términos generales Visual Basic es una plataforma de desarrollo de aplicaciones.

Visual Basic proporciona elementos gráficos de diseño de aplicaciones, compilador, editor de programas con capacidad de depuración al instante y asistencia basada en colores, ayuda contextual, poderosos asistentes que automatizan tareas de uso común, tan disímolas en su objeto como en su complejidad, por citar sólo algunas características.

A continuación se presenta la interfaz gráfica desarrollada para la implementación del controlador difuso.

Cuando se abre el programa, en la barra de herramientas aparecen los botones siguientes:



Figura 2.1. Barra de herramientas principal.

En la figura 2.1 se presentan los ocho botones presentes en la pantalla principal:



Agrega un bloque de reglas difusas



Agrega una entrada difusa



Agrega una salida difusa



Abre la terminal de comunicación



Generar código para MCU



Generar archivo ASM



Guardar archivo ASM



Descargar archivo a MCU



Envía comando de ejecución del programa previamente descargado al MCU, localizado a partir de la memoria 800h. (solo funciona si el puerto serie esta abierto)



Detiene la ejecución del programa del controlador difuso. (solo funciona si el puerto serie esta abierto)



Abrir adquisidor de datos

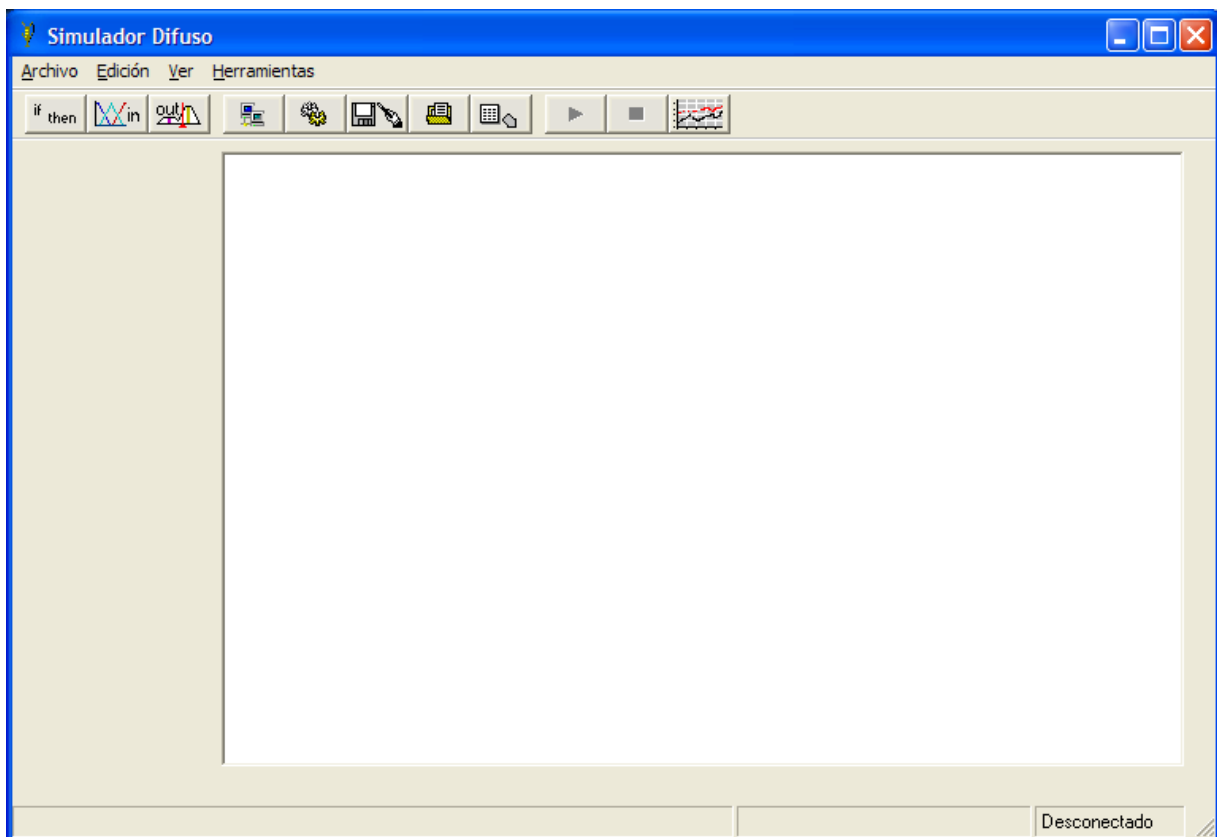


FIGURA 2.2. Pantalla de inicio de la interfaz gráfica de la aplicación de desarrollo para controladores difusos.

En la figura 2.2 se observa la pantalla de inicio donde la primera opción del menú Archivo es la de un programa nuevo, en la parte derecha inferior de la pantalla se observa un cuadro de Status de la aplicación y es mediante este cuadro de Texto que se muestra la conexión y desconexión del MCU con la aplicación. Cabe mencionar que se agregó la característica de determinar si el puerto de comunicación está disponible cuando se selecciona de la barra de menú la opción Herramientas y luego Preferencias, apareciendo la pantalla mostrada en la figura 2.3, en la cual puede elegirse tanto el puerto a utilizar así como la velocidad de la conexión.



FIGURA 2.3. Pantalla de configuración del puerto de comunicación.

Cuando se abre la opción del menú principal Archivo se abre la ventana que se muestra en la figura 2.4.

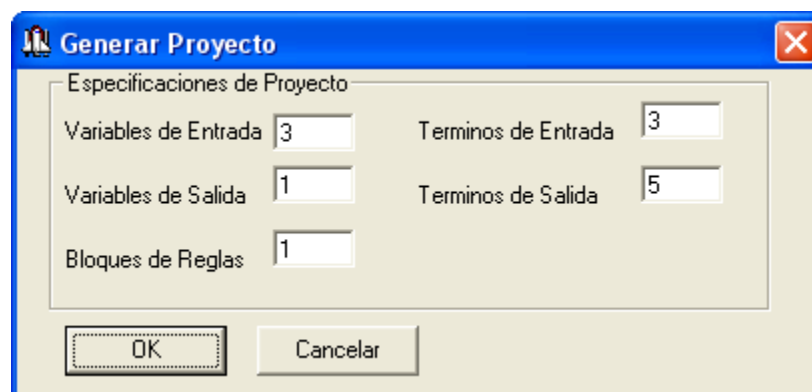


FIGURA 2.4. Pantalla de definición general del proyecto, para la creación de forma rápida de un controlador difuso.

En la figura 2.4 se muestra la ventana desde la cual se puede generar un proyecto, en esta se define la cantidad de entradas, la cantidad de salidas así como los bloques de reglas.

Al presionar el botón Ok se genera el proyecto en la pantalla de desarrollo, si no se cambian, se eligen los valores por defecto de las propiedades del programa a desarrollar,

se genera un proyecto con tres entradas, un bloque de reglas y una salida. El resultado se observa en la figura 2.5, más las propiedades de cada término de tanto de entrada, salida y la generación de reglas contenidas en el bloque de reglas deben hacerse paso a paso por el usuario.

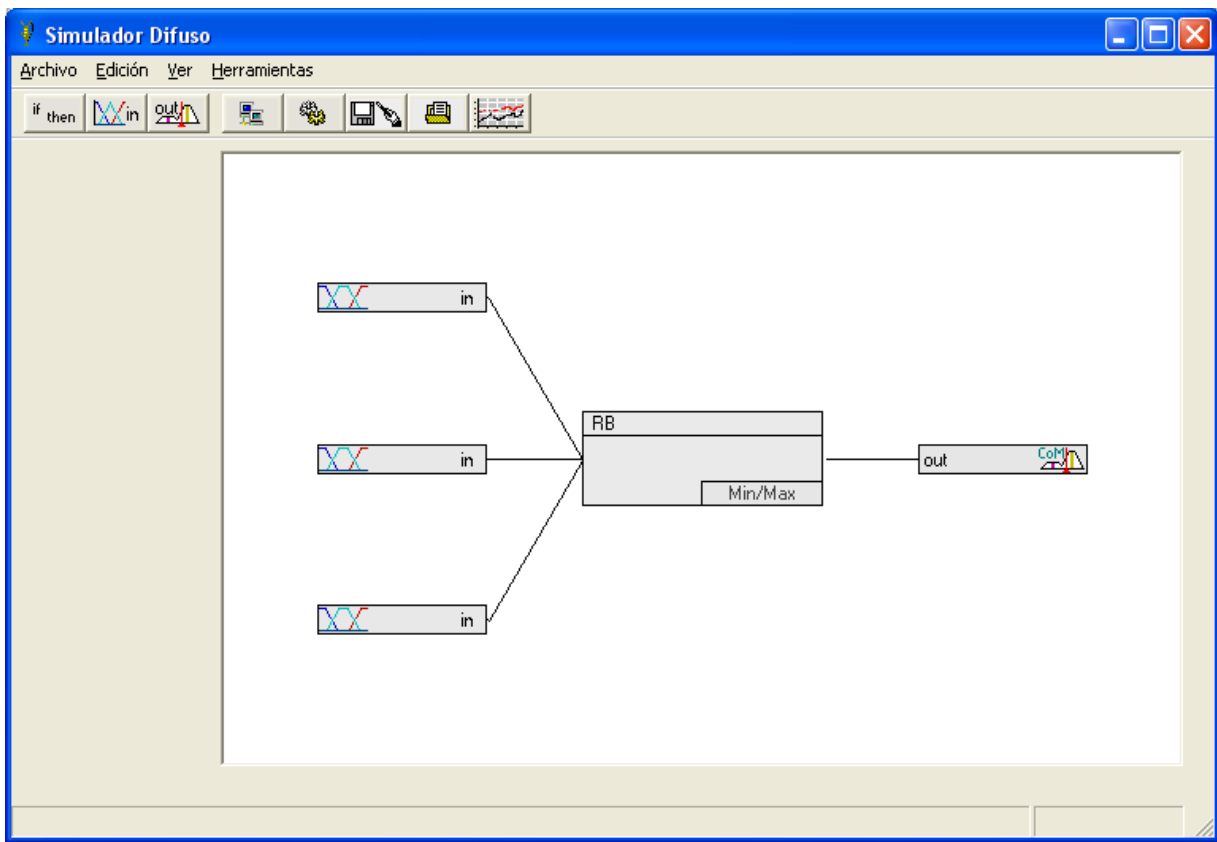


FIGURA 2.5. Pantalla de desarrollo gráfico del controlador difuso.

Cuando se posiciona el cursor sobre la representación gráfica de la entrada y se hace clic derecho se tiene acceso a los parámetros de configuración de la entrada respectiva, en esta pantalla se puede configurar la cantidad de términos que tendrá la entrada así como la forma de éstos. Dentro de las posibles formas están la triangular y la trapezoidal, teniéndose la posibilidad de modificar los puntos que representan a cada una de las aristas

de los términos, lográndose así modificar la intersección de estos con los ejes y el ancho de los mismos.

En la parte inferior izquierda se puede configurar los valores para el Universo de discurso, el cual definirá las cotas del eje de las abscisas. Más debe tenerse en cuenta que ello es únicamente una herramienta para una mejor y más fácil comprensión del proceso a controlar, por que los valores que puede el sistema tomar como entradas en forma discreta varían siempre de 0 a 255, y es sobre esta base como se generan los conjuntos difusos que posteriormente serán usados para el cálculo de la salida del sistema; el simulador difuso en virtud de la característica antes mencionada se encarga de calcular y escalear los conjuntos difusos definidos por el usuario, así por ejemplo si se tiene una variable temperatura que posee un discurso de 15°C a 150°C, el usuario puede establecer dichos límites en la pantalla correspondiente a la variable temperatura y sobre esa base definir los conjuntos difusos de su sistema, trabajando cada uno de ellos en la unidades °C de temperatura, y el simulador difuso se tomará la tarea de hacer el respectivo escaleamiento sobre la base de un discurso de 0 a 255 pasos de la variable de entrada y sobre ello calculará los respectivos conjuntos difusos en magnitudes comprensibles para el MCU68HC12.

El botón de aceptar además de cerrar la ventana de definiciones de la entrada que se está editando, se encarga también de actualizar los cambios en la variable en edición y pasa dicha información a las variables globales del sistema, para que estas sean capaces de realizar los cálculos respectivos ya sea para llevar a cabo la depuración off line como para la generación del archivo de código ASM compilable. Puede observarse un ejemplo de una variable sin editar en la figura 2.6.

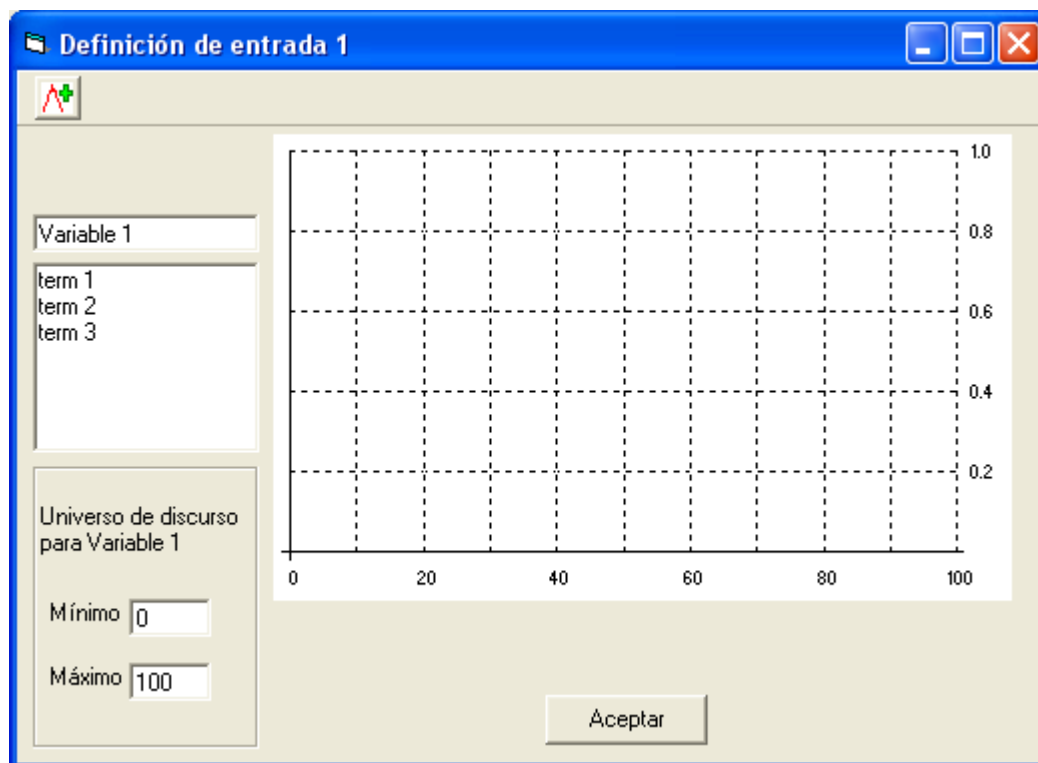

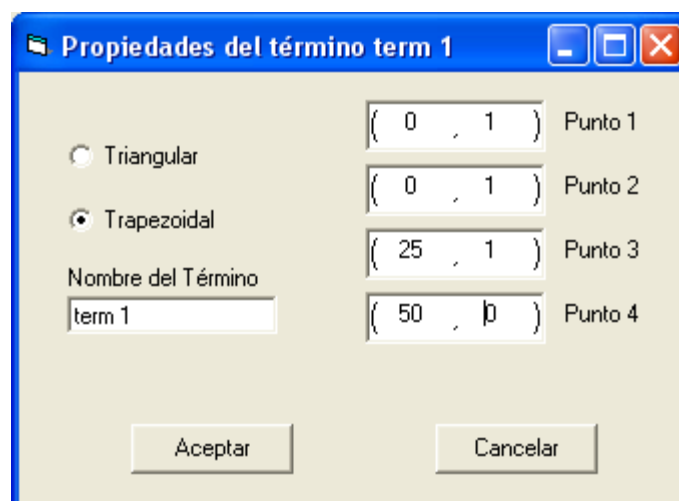


FIGURA 2.6. Pantalla de definición de la entrada 1. Definición de variable a utilizar y sus términos.

Para editar las propiedades y los valores de los puntos que componen la gráfica del término se hace doble clic en el nombre del término, el cual genéricamente aparece como term1, term2, term5. Por defecto se ha definido que cada variable tendrá por lo menos 3 términos, en caso de no usarse uno o más de ellos simplemente puede obviarse su utilización durante la definición del bloque de reglas, más es imperativo que este se defina aunque sea con propiedades cualquiera, solamente que para obviar su utilización durante la generación de las reglas se debe estar seguro que este representa condiciones que bajo ninguna circunstancia pueden darse a la entrada de un sistema, si se considera

que las mismas pueden existir aun cuando se trate de condiciones de error del sistema, deberá considerarse en el bloque de reglas su utilización para que bajo dichas condiciones no se ocasionen daños al proceso, a la planta, actuador, incluso a los seres humanos que pudieren estar en o cerca del sistema. Por otra parte, para agregar un termino o conjunto a cada entrada (máximo de términos permitido es cinco), debe presionarse el botón  de la barra de herramientas, si se desea asignar el nombre a la variable, bastara con escribir directamente sobre el cuadro de texto en la esquina superior izquierda de la ventana de propiedades de la variable a editar y este automáticamente se actualiza para los demás procesos que hagan referencia a ella; la retícula que aparece en la pantalla es una ayuda para la representación gráfica de los conjuntos difusos sobre su universo de discurso y es sobre la misma que al momento de ejecutar la depuración off line donde aparece una barra que resalta el nivel de pertenencia de una variable a los conjuntos difusos definidos, según esta realiza una excursión sobre el universo de discurso fijado.

Primero es necesario hacer doble clic sobre el término 1 en cuadro de lista que aparece en la esquina superior izquierda de la ventana de propiedades de la entrada, y aparecerá la ventana que se muestra en la figura 2.7.



Propiedades del término term 1

☐ Triangular
☒ Trapezoidal

Nombre del Término: term 1

(0 , 1)	Punto 1
(0 , 1)	Punto 2
(25 , 1)	Punto 3
(50 , 0)	Punto 4

Aceptar Cancelar

FIGURA 2.7. Pantalla de propiedades del término 1 en la cual se determinan los puntos de la gráfica que define la función de membresía. Definido por defecto como trapezoidal.

Debido a que el termino 1 por defecto se ha definido como trapezoidal, no se hace necesario seleccionar esta opción, por lo que se procede a escribir directamente sobre los cuadros de texto que contienen los puntos respectivos de cada arista, para el caso de los términos 1 y 3 cuando estos sean definidos como trapezoidales en los puntos de la arista más a la izquierda para el termino 1 y la arista más a la derecha se ha restringido el valor de su abcisa a ser igual al limite inferior del universo de discurso y al limite superior del mismo para cada caso, y así mantener una congruencia con las condiciones establecidas en la variable de entrada bajo edición, así para estos dos casos particulares el usuario tiene acceso a modificar el valor que tendrá el conjunto cuando este alcance Los limites superior e inferior del universo de discurso, y, los cuales pueden ser cero o uno, las aristas centrales indistintamente si el termino es triangular o trapezoidal siempre podrán ser definidos los valores de sus abscisas, ya que para el caso de las ordenadas es imperativo que estas siempre tengan el valor de uno y evitar así la definición impropia de los conjuntos difusos. Los valores que aparecen por defecto para el termino 1, son los presentados en la figura 2.8, en el caso que se desee usarlos tanto para este término como el resto, bastara con presionar aceptar para que los conjuntos queden definidos para los cálculos posteriores, y así cada vez que se ingrese a dicha ventana se actualizaran las propiedades si en caso que haya ocurrido algún cambio.

De la misma manera, como se muestra en la figura 2.8 se pueden editar los valores para el término 2, el cual por defecto se define como triangular con los valores que se presentan a continuación:

Propiedades del Término 2

☒ Triangular
☐ Trapezoidal

Nombre del Término:

Punto 1: (25 , 0)
 Punto 2: (50 , 1)
 Punto 3: (75 , 0)
 Punto 4: (0 , 0)

FIGURA 2.8. Pantalla de propiedades del término 2 en la cual se determinan los puntos de la gráfica que define la función de membresía. Definido por defecto como triangular.

Tanto para el término 2 como el 4 y 5 se han definido por defecto como triangulares, más el usuario puede redefinirlos como trapezoidales, en el caso de estos mismos, puede modificarse únicamente los valores de las abscisas, siempre teniendo en cuenta que las mismas no sobrepasen los límites establecidos en el universo de discurso, ello generará un mensaje de error y no aceptará los datos introducidos mientras no se corrija el error.

El término tres se ha definido por defecto como trapezoidal, mostrado en la figura 2.9 y los puntos asignados por defecto son los siguientes:

Propiedades del Término 3

☐ Triangular
☒ Trapezoidal

Nombre del Término:

Punto 1: (50 , 0)
 Punto 2: (75 , 1)
 Punto 3: (100 , 1)
 Punto 4: (100 , 1)

Figura 2.9. Pantalla de propiedades del término 3 en la cual se determinan los puntos de la gráfica que define la función de membresía. Definido por defecto como trapezoidal.

Este término comparte la característica del término 1 de ser el que esta en uno de los límites del universo de discurso por lo mismo y como ya se explico detalladamente con anterioridad, la edición de la abcisa de la arista más a la derecha esta restringido al usuario, el valor de esta depende únicamente del limite superior del universo de discurso, quedando únicamente a disposición del usuario la edición de la ordenada respectiva como puede apreciarse en la figura 2.10. y al igual que para los términos anteriores, pueden usarse los valores por defecto de este con solo presionar aceptar, los mismos pueden ser modificados en cualquier momento por el usuario, teniendo en cuenta que si se elige cancelar cualquier modificación que se haya hecho quedará sin efecto para todos los procesos y cálculos posteriores.

Una vez definidas las propiedades de los términos que componen una entrada, se puede obtener una representación gráfica como se presenta la figura 2.11. Es de notar que para los tres conjuntos difusos que componen esta entrada se han usado los valores por defecto, como ilustración, sin embargo tanto el nombre de la variable, los términos y el universo de discurso para la correspondiente entrada puede modificarse en el momento que se desee, más se aclara que para el caso que los conjuntos difusos hayan sido dibujados ya sea con los valores por defecto u otros definidos por el usuario, y se realice una modificación al universo de discurso de la entrada, los mismos no se modificarán en la pantalla, hace falta seleccionar uno a uno los términos y presionar únicamente el botón aceptar para que sean estos redibujados sobre la cuadrícula presentada.

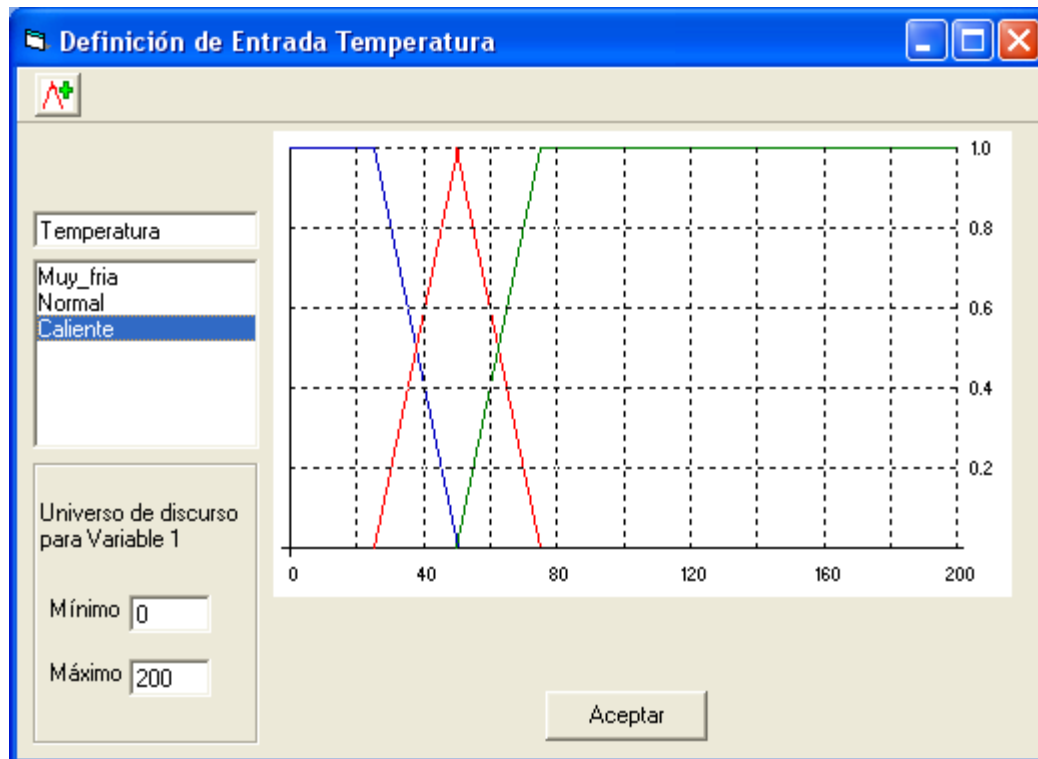


Figura 2.10. Ventana de edición de las propiedades de la variable Temperatura definida sobre un universo de discurso de 0 a 200 grados.

En la figura 2.10 se presenta un ejemplo de lo mencionado anteriormente respecto a la modificación del universo de entrada, como se puede apreciar los conjuntos difusos siguen con los valores por defecto, habiendo cambiado únicamente la representación gráfica de los mismos en función del universo de discurso. De no abrir las ventanas de propiedades de los términos uno a uno aparecería una representación como la mostrada en la figura 48, que aparentemente indicaría que el conjunto 2 tiene su arista central en la posición (100,1), y así sucesivamente para los demás términos.

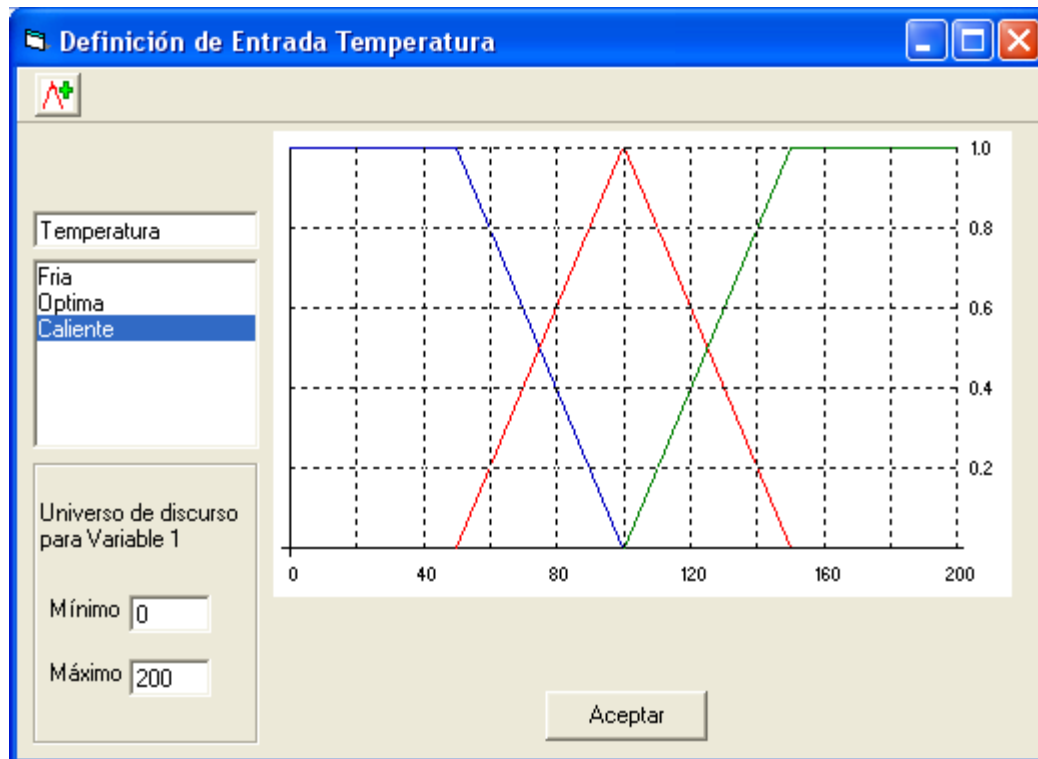


Figura 2.11. Pantalla representativa de las funciones de membresía para la entrada 1 con datos de usuario.

Es importante en lo referente a los nombres de los términos de entrada como de salida, que estos no sean repetidos, o en el caso que sus nombres estén formados de más de una palabra, que no exista espacios en blanco entre ellas como para el caso mostrado en la figura 2.11, con el término “Muy_fria” que lleva un guión bajo intercalado entre ambas palabras; si bien es cierto que para el simulador difuso estas restricciones no son importantes debido a la complejidad del lenguaje de programación, más sin embargo para cuando sea generado un archivo con código ASM compilable, ello ocasionará errores de compilación, pues el ensamblador no acepta espacios en blanco en la definición de términos, ni nombres repetidos de los mismos.

En el caso de las salidas se posiciona el puntero sobre la representación gráfica de la salida y se hace clic derecho para abrir la ventana de edición de sus propiedades, como se muestra en la figura 2.12.

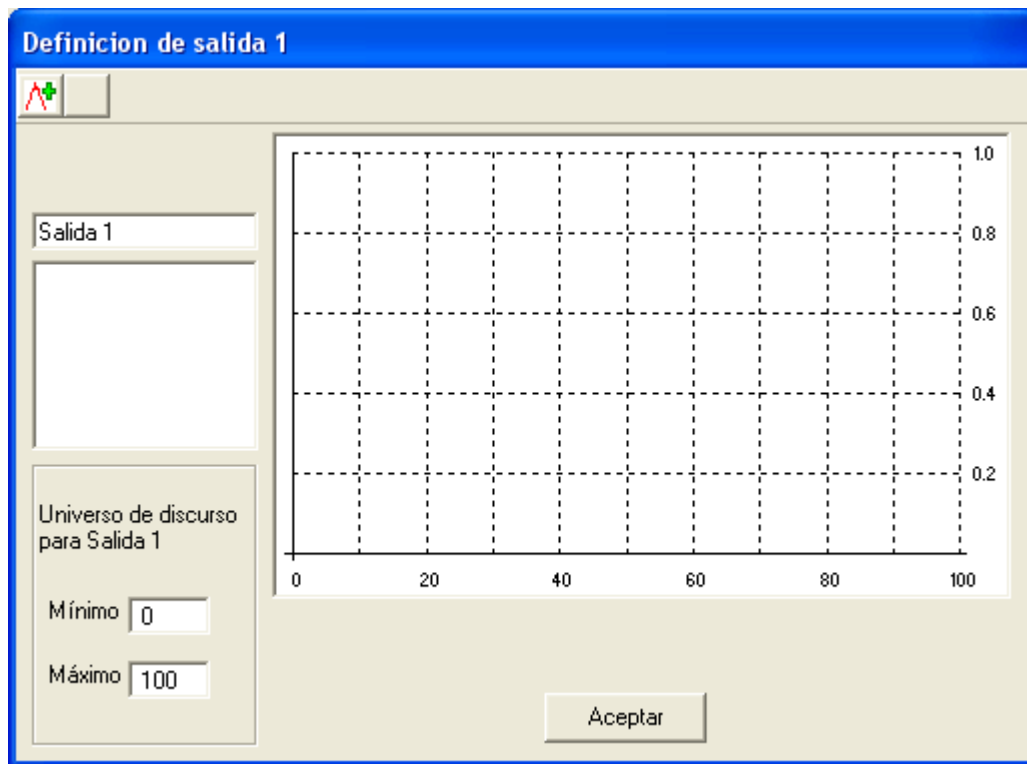



Figura 2.12. Pantalla de definición de salida

Debe definirse por lo menos una salida para el sistema, por medio del botón  en la barra de herramientas se puede agregar una salida al sistema.

Cuando se presiona el botón de agregar salida se puede configurar la salida, según la figura 2.13.

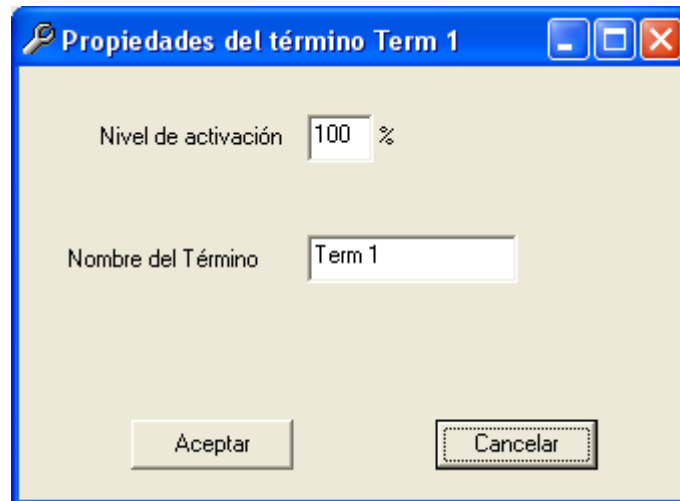


Figura 2.13. Definición de salida

En la figura 2.13 se observa los cambios a la configuración que pueden realizarse en las salidas: su nivel de activación que corresponde a la “fuerza” que se le da a dicho termino de entrada, en función de porcentaje y el nombre del término que es el identificador del mismo en el simulador difuso y lo será en archivo de código ASM compilable, al igual que las entradas, los términos de las salidas no deben tener nombres iguales entre si, tampoco iguales a otro término de cualquiera otra entrada, para ello el programa realiza una verificación de dicha situación previo a desplegar la ventana de edición del bloque de reglas, en caso de encontrar duplicados, dará el mensaje de error que se presenta en la figura 2.14. Y ante la ocurrencia de este caso es preciso buscar con detalle dentro de las ventanas edición de propiedades de las entradas como de la salida, ya que la duplicidad de nombres ocasiona errores indistintamente entre las entradas o entre entradas y salidas. Puede observarse de la figura siguiente que el error se debe a la presencia del término “dos” en las entradas Variable_1 y Variable_2, motivo por el cual no se permitirá el acceso a la edición del bloque de reglas hasta eliminar dicha situación con el fin de evitar la generación de reglas con valores erróneos para el compilador ASM, del MCU 68HC12

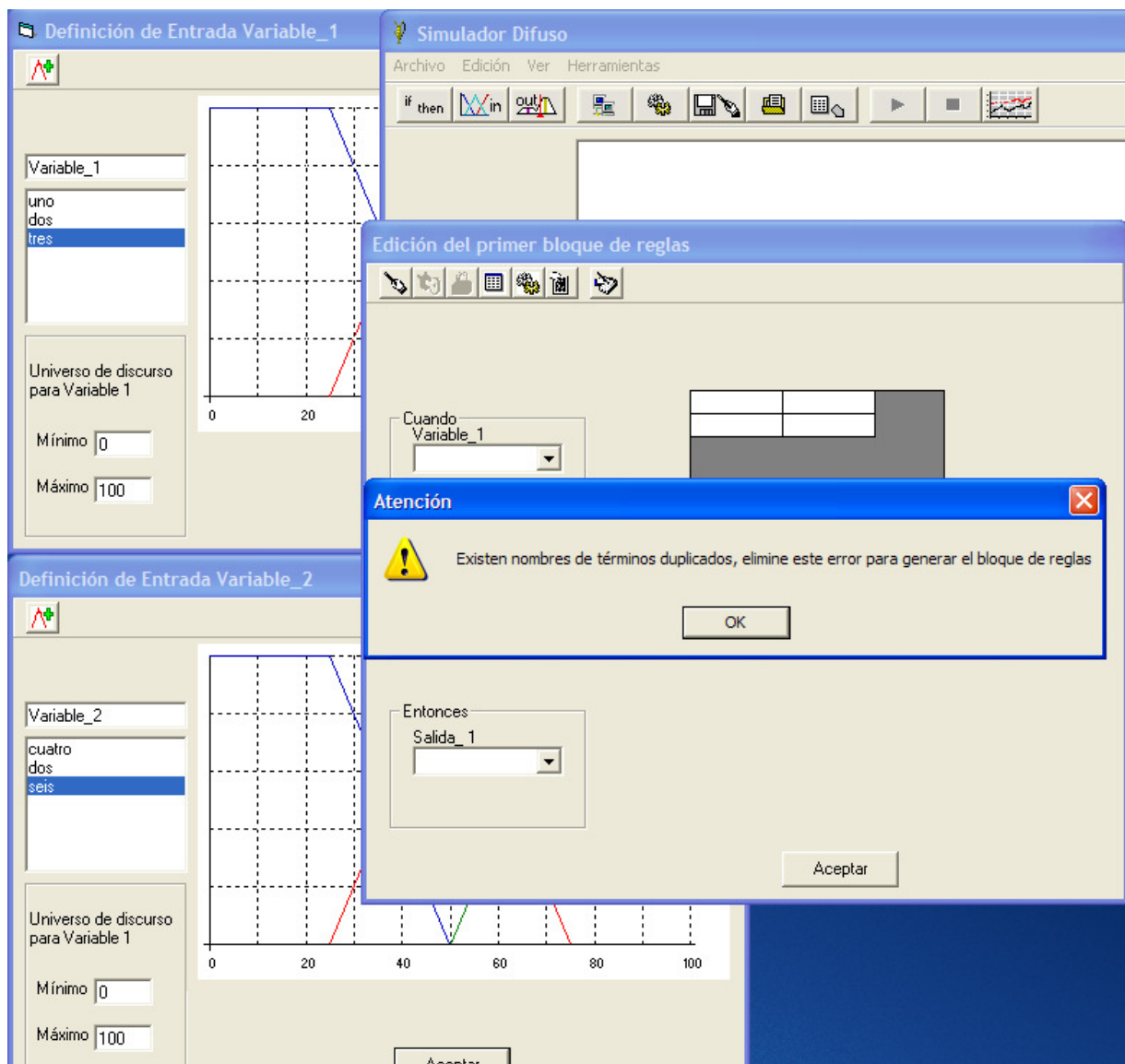


Figura 2.14. Ejemplo del mensaje de error desplegado en el caso que el programa principal encuentre nombres duplicados, en este caso el término “dos” se repite en la entrada 1 y 2

Así mismo debe evitarse el uso de espacios en blanco, en el caso que el usuario los escriba simplemente el programa los sustituirá por un guión bajo “_” sin solicitar confirmación al usuario, todo esto se hace con el fin de evitar posibles errores de sintaxis en la compilación del archivo ASM que se vaya a generar, derivado del controlador difuso definido por el usuario. No obstante se aclara que ello no implica que las reglas

sean correctas, pues dependerá exclusivamente del usuario la elección apropiada de las mismas.

La figura 2.15 muestra el ejemplo de una salida denominada “Válvula” con dos términos tipo barra (singleton), con un nivel de activación de 10% para el término “Cerrada” y de 100% para el término “Abierta”

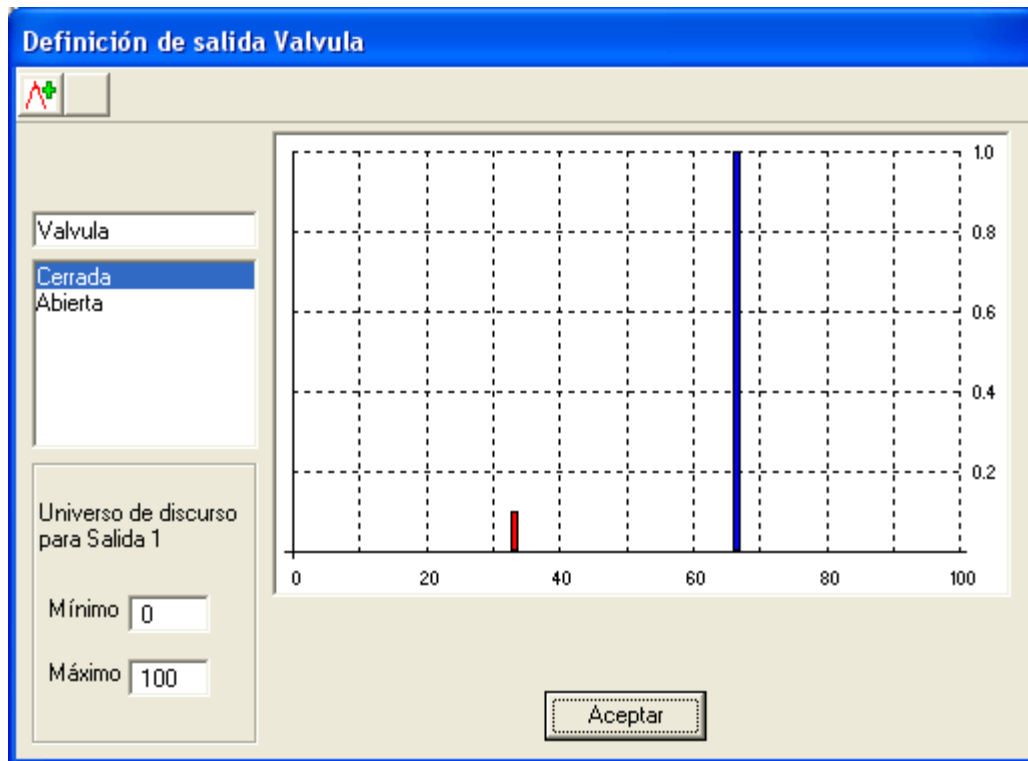


Figura 2.15. Representación de salidas tipo singleton, la de la izquierda con un nivel de activación de 10% y la de la derecha con 100% de activación.

Una vez editadas tanto entradas como salidas, se procede a abrir el bloque de reglas para generar las mismas, no es necesario que sea en ese orden pero para que tanto las reglas como los nombres de los términos tengan un significado menos abstracto para el usuario, es importante que se realicen los pasos en el orden detallado anteriormente, en el bloque de reglas que se muestra en la figura 2.16, puede verse una matriz cuyas filas y columnas tienen los nombres de los términos de las entradas, en el caso que sea una sola variable la

matriz tendrá una sola columna y tantas filas como términos tenga la variable de entrada, para el caso de dos variables de entrada, la matriz tendrá tantas columnas como términos tenga la variable de entrada 2, y tantas filas como términos tenga la variable de entrada 1; Aparecerán cajas de texto desplegable con listas de opciones para cada entrada y para la salida.

Edición del primer bloque de reglas

Quando

Temperatura

Caliente

Humedad

Alta

Entonces

Ventilador

Rapido

Aceptar

Temperatura

Fria	Lento	Lento	Lento
Optima	Lento	Medio	Medio
Caliente	Medio	Medio	Rapido



Baja Normal Alta

Humedad

Figura 2.16. Definición de las reglas difusas


Para que por medio de la matriz de reglas se pueda generar el listado de las reglas de inferencia difusa, basta con seleccionar el término deseado de la salida y con el mouse hacer clic sobre la casilla donde se intercepten el termino de la variable 1 y el respectivo de la variable 2 deseados, en caso de un error basta hacer doble clic sobre la casilla para

que ésta quede en blanco o se puede seleccionar el término correcto y se hace clic sobre el erróneo.

Para que se generen las reglas de acuerdo a las relaciones establecidas en la matriz del bloque de reglas, se hace clic sobre el botón de la barra de herramientas , y para verlas en forma de reglas IF-THEN debe hacerse clic sobre el botón , y aparecerá un cuadro como el que se presenta en la figura 2.17.

Vista de reglas		
SI		ENTONCES
Temperatura	Humedad	Ventilador
Fria	Baja	Lento
Fria	Normal	Lento
Fria	Alta	Lento
Optima	Baja	Lento
Optima	Normal	Medio
Optima	Alta	Medio
Caliente	Baja	Medio
Caliente	Normal	Medio
Caliente	Alta	Rapido
Aceptar		

Figura 2.17. Definición de las reglas difusas

Una vez generadas las reglas en formato texto se pueden depurar el controlador y sus reglas por medio del botón  de la barra de herramientas de la pantalla de configuración de las reglas difusas, el cual despliega la ventana mostrada en la figura 2.18 donde aparecen barras deslizables cuyo valor se convierte en la entrada puntual para cada una de las variables del sistema, y, que serán alimentados a las reglas de inferencia

difusa, para obtener así una respuesta a las condiciones de entrada establecidas por el usuario, de acuerdo al controlador previamente diseñado, este proceso es de extrema utilidad, ya que permite depurar el controlador y de esta forma verificar todas las condiciones de entrada posibles aún aquellas consideradas imposibles de ocurrir en el mundo físico bajo condiciones extremas; sin la necesidad de llegar al paso de la implementación, generación de código ejecutable, uso de transductores (los cuales dependiendo del fenómeno que midan por ejemplo temperatura puede tomar mucho tiempo el evaluar las condiciones posibles de entrada), incluso es de gran utilidad en la determinación de los conjuntos difusos, ya que toda la base de control que el sistema tendrá será determinada por la correcta definición de los mismos, proceso que de realizarse por cálculos manuales o mediante hojas de cálculo electrónicas puede ser un poco tedioso debido a la definición de las funciones que determinan a los mismos.

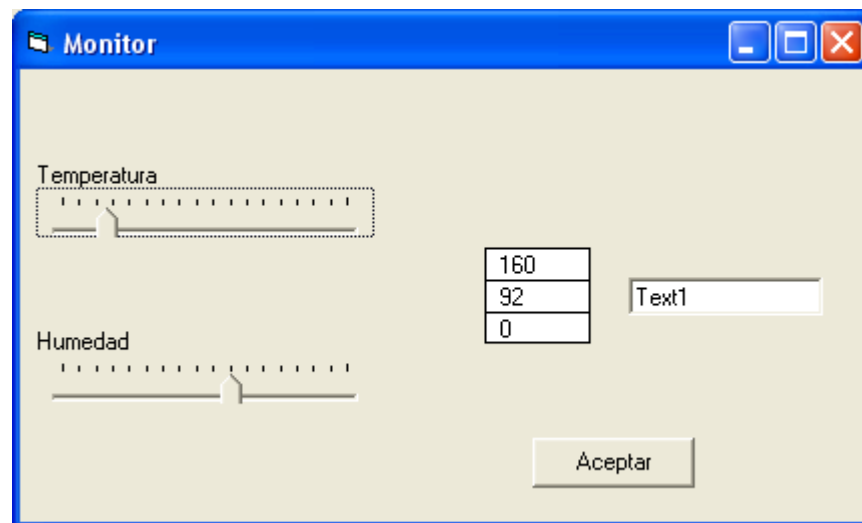








Figura 2.18. Ventana de depuración off line

En la ventana monitor, mostrada en la figura 2.18, también se presenta un vector columna⁷ en el cual se puede apreciar los términos de la salida, a su vez puede observarse

⁷ Representación en una sola dimensión vertical de datos

como cada uno de estos varía de acuerdo a las entradas del sistema y la influencia de los mismos sobre la salida defuzificada que se presenta en el cuadro de texto.

Una vez se haya definido correctamente los nombres de las variables de entrada como sus respectivos términos, de igual manera para la salida, es importante realizar la depuración detallada previamente, pues del estudio de las gráficas que se obtengan de la simulación se podrá determinar si el sistema responde de la forma deseada sobre la base de las condiciones que puedan presentarse a la entrada; paso seguido de la simulación y análisis de la respuesta del controlador pueden usarse los botones de la barra de herramientas para generar el archivo ASM correspondiente , luego guardar el mismo con un nombre que nos permita identificarlo en el directorio de trabajo con el botón , es importante que se realice este paso ya que el anterior genera un archivo de trabajo temporal sin extensión ASM, por lo cual se debe guardar el mismo con extensión .ASM para el paso siguiente que corresponde a la compilación con el botón , proceso que necesita se le defina un archivo con dicha extensión, caso contrario obviara la compilación; obtenido el archivo .S19 derivado del paso anterior se procede a abrir el terminal de comunicaciones, para determinar si este está abierto o no, puede observarse en la figura 2.19 en la esquina inferior derecha que presenta el estado actual del terminal. Una vez el estado de este sea conectado los botones   estarán activados, y solo entonces se puede proceder a descargar el archivo .S19 al MCU mediante el botón , que abrirá una ventana que le permite al usuario seleccionar el archivo a descargar el archivo.S19 al MCU vía el puerto de comunicaciones serie.

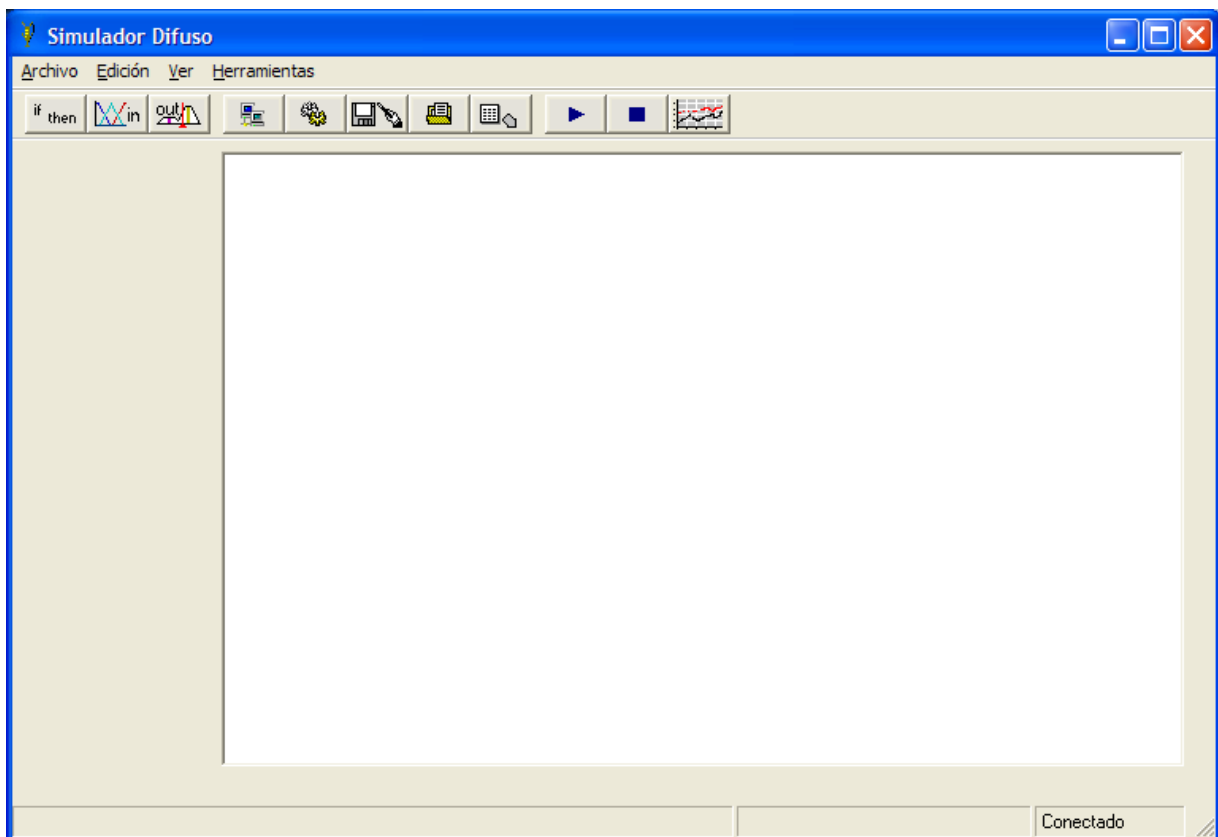




Figura 2.19. Pantalla del simulador difuso cuando el puerto serie esta conectado

En la figura 2.20 se observa que luego de presionar una vez el botón de inicio  se desactiva quedando como opción la de detener el programa con el botón , el cual no se desactiva a menos que el usuario cierre el terminal de comunicaciones, la razón es que existe la posibilidad que el usuario desconecte la PC del microcontrolador una vez lo haya puesto en marcha, el mismo puede funcionar como un sistema “stand alone” donde no necesita de ningún tipo de monitoreo, ni depende de ninguna señal proveniente de una PC u otro MCU, así cuando se desee conectar de nuevo la PC para graficar su salida o para detener el proceso en ejecución actualmente por mantenimiento, mejoras o cambio del proceso estará disponible la opción.

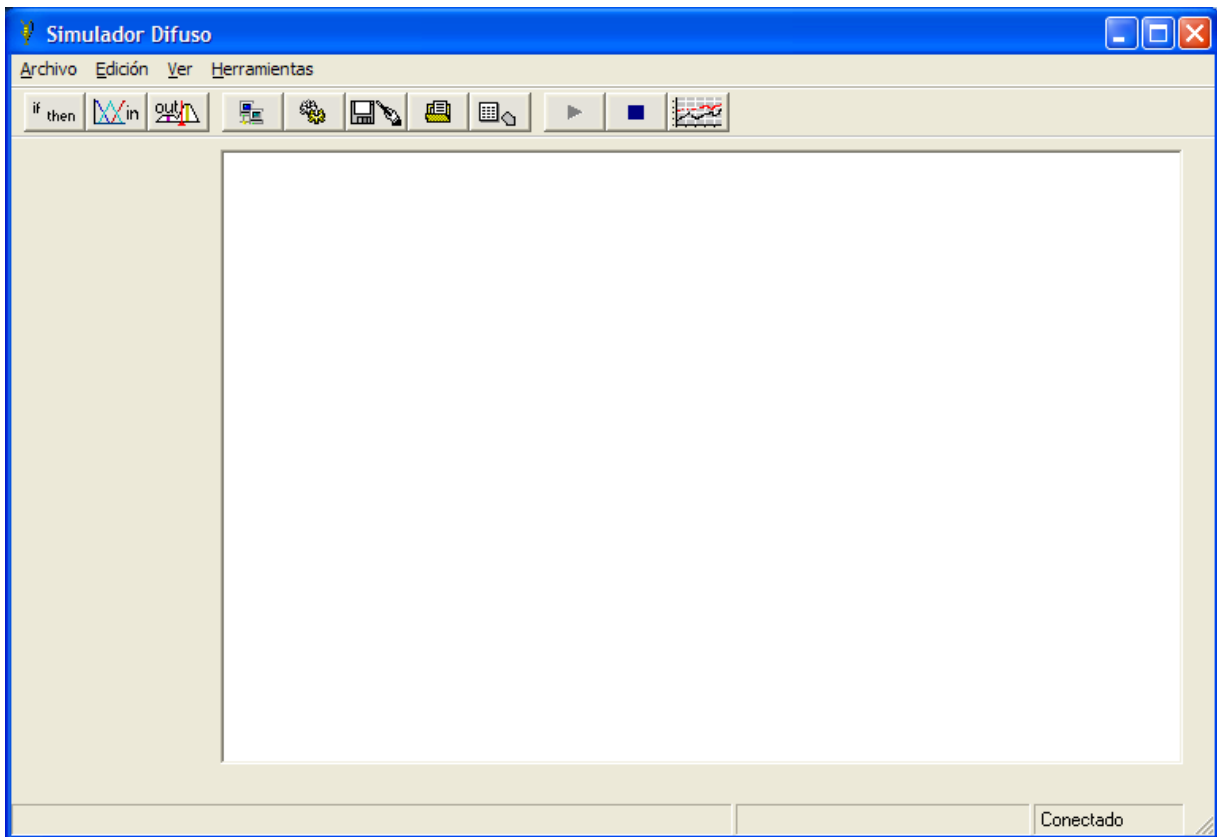


Figura 2.20. Simulador difuso luego de enviar comando de ejecución del programa residente en el MCU

Cada vez que el usuario presione el botón de interrupción de la ejecución del programa en el MCU, se presentará un mensaje de confirmación para que el usuario detenga la ejecución, ya que en el caso de encontrarse detenido el MCU y presionar dicho botón primero para luego enviar el comando de ejecución, en la secuencia mencionada podría causar un comando invalido, y con el fin de evitar tal situación se solicita la confirmación por parte del usuario como se muestra en la figura 2.21.

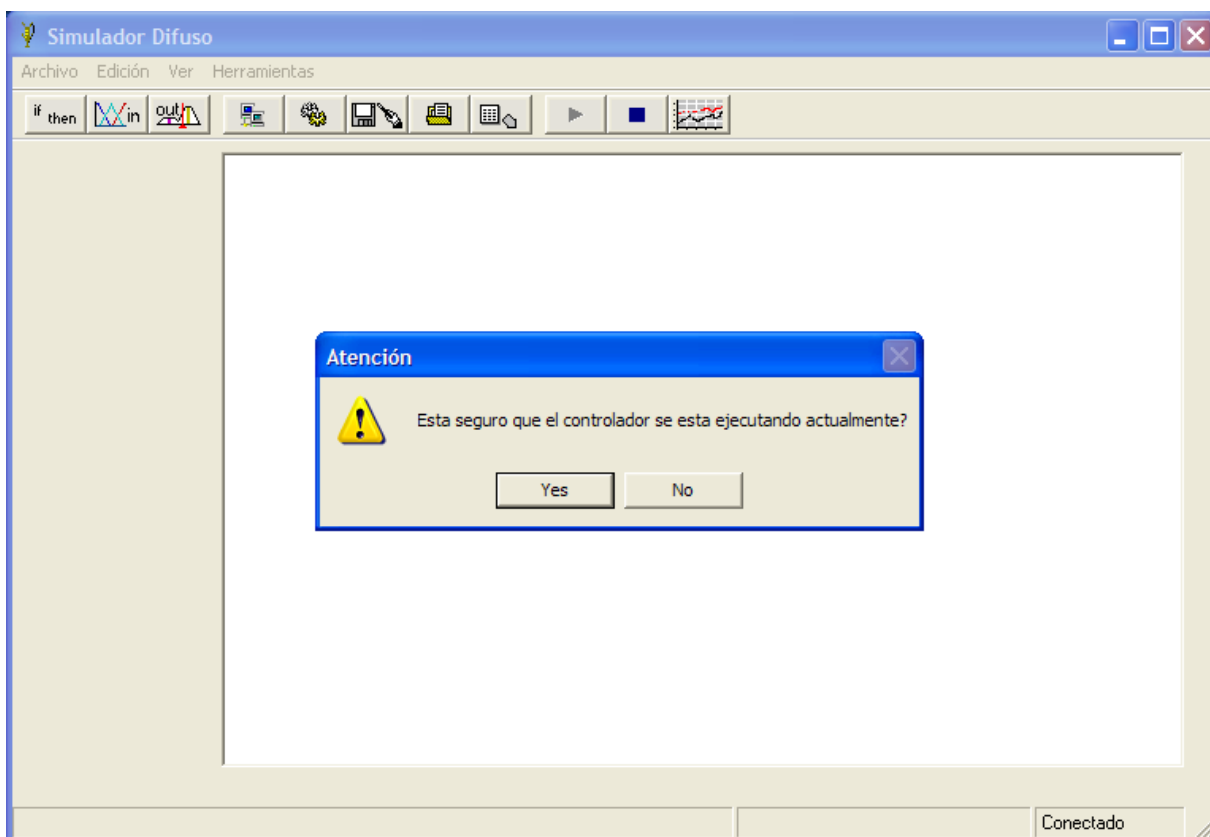


Figura 2.21. Mensaje de solicitud de confirmación para detener la ejecución del MCU

RESULTADOS OBTENIDOS CON EL CONTROLADOR DIFUSO

A continuación se presentan los diferentes experimentos realizados tanto con el sistema de control de nivel de líquido, velocidad en un motor eléctrico y temperatura en una centrifugadora de aceite lubricante⁸; en los primeros dos casos se experimentó con el sistema a lazo abierto y luego se procedió a implementarlos con un control proporcional; dichos experimentos fueron realizados en las instalaciones del laboratorio de Instrumentación y Control Automático del Centro de Investigaciones y Transferencia de Tecnología (CITT) de la Universidad Don Bosco. Para el caso de la centrifugadora de aceite, únicamente se experimentó con el sistema trabajando bajo la acción de control de un sistema Proporcional Integral.

2.1 SISTEMA DE CONTROL DE NIVEL DE LIQUIDO

2.1.1 Nivel de Líquido a lazo abierto

En la figura 3.1 puede apreciarse la gráfica del sistema de nivel de líquido a lazo abierto, la razón de realizar dicho experimento como en todo proceso que se desee controlar usando Lógica Difusa, es para determinar las características del mismo, que para este caso son los niveles inferior y superior del líquido, así como la velocidad de llenado con un caudal de escape, equivalente al gasto volumétrico para un proceso determinado (dicho gasto no está definido, más pudiera considerarse también como una variable de

⁸ Experimento realizado en una centrifugadora Alfa Laval en las instalaciones de la generadora de energía eléctrica Nejapa Power Plant

entrada del sistema, para el caso no se contaba con ningún proceso donde pudiera aplicarse tal situación); y, con base a lo anterior definir los conjuntos difusos del sistema, los cuales se presentarán más adelante. Para implementar este proceso se contó con un transductor de Presión, el cual hace uso de un tubo a forma de sonda que se insertó en el tanque y cuya columna de aire ejerce una presión sobre el transductor proporcional a la columna de agua dentro del tanque, ello fue posible ya que se trabajó con un líquido a temperatura constante.

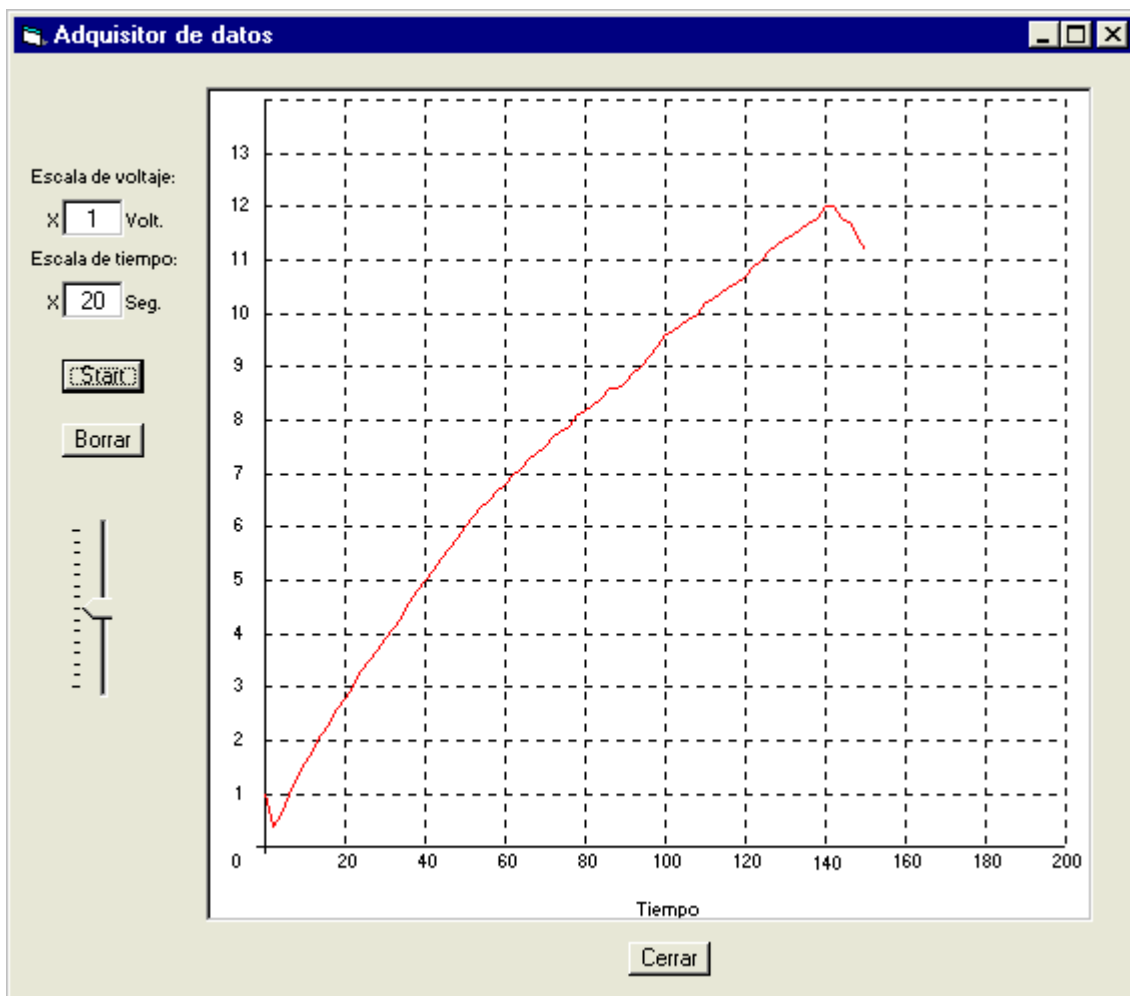


Figura 3.1. Gráfica de la respuesta a lazo abierto para el control de nivel

3.1.2 Nivel de líquido a Lazo Cerrado, y Controlador Proporcional

Para la planta de Control de Nivel, se utilizó el controlador Proporcional Lucas Nulle existente en el Laboratorio de Instrumentación y Control del CITT, la gráfica fue obtenida con el software de Adquisición de Datos⁹. En este experimento, en donde el seteo de nivel estaba al 75% del máximo nivel en el tanque, con K_p de 2.5, de lo observado en la gráfica de la figura 57 puede decirse sobre el sistema lo siguiente: (a) no presenta sobreimpulso y (b) el tiempo de estabilización del sistema es de 50 segundos aproximadamente.

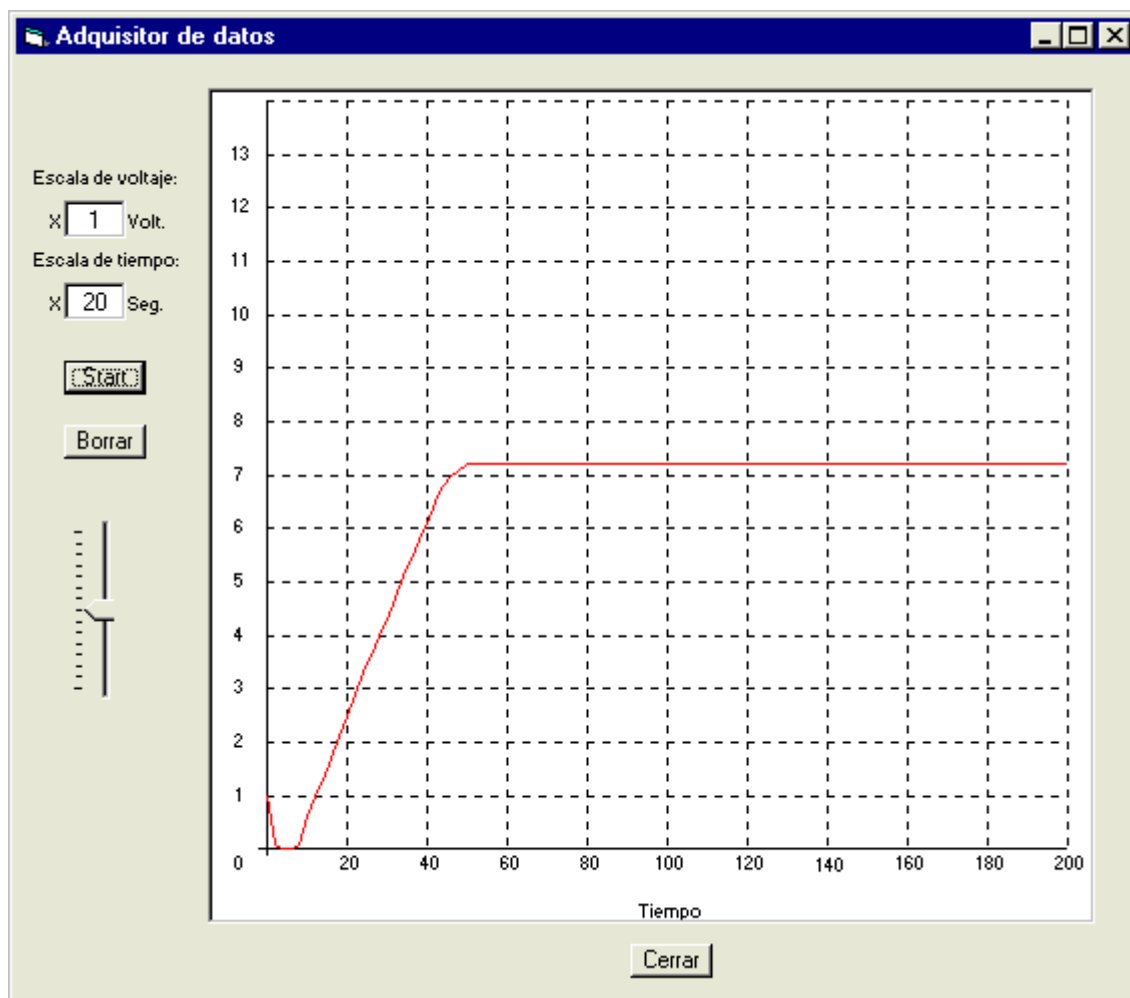


Figura 3.2. Representación gráfica de un controlador P para nivel

⁹ Aplicación desarrollada durante el trabajo de graduación . Presentado en el Apéndice A3

3.1.3 Implementación del controlador difuso

De las características observadas en la gráfica de lazo abierto, así como del funcionamiento a lazo cerrado con un controlador proporcional, se determinó que el Sistema de Nivel de Líquido requiere de dos variables de entrada (sustituyendo la variable error) la primera denominada “setpoint” o valor de referencia que para el caso varía según el usuario decida desde vacío hasta completamente lleno, así mismo se hace necesario considerar la variable “floop” que corresponde al lazo de retroalimentación; En las situaciones donde el punto de referencia sea variable como es el caso puede usarse una variable error que puede ser alimentada al controlador considerando que este no acepta valores negativos de voltaje aún cuando los pines V_{RH} y V_{RL} , estén conectados a +2.5V y -2.5V respectivamente, ya que según las características del convertidor incorporado al MCU ignora los valores de referencia negativos; Para lo cual se hace necesario de Hardware adicional que acondicione dicha señal de 0V a 5V.

Por lo antes mencionado se consideró óptimo de acuerdo a los recursos disponibles con el microcontrolador hacer uso de las variables antes mencionadas; definiendo este punto se procedió a determinar los conjuntos necesarios para el controlador difuso a implementar, si bien es cierto se consideraron inicialmente tres (vacío, medio y lleno), pero los mismos no eran suficientes para tener un control comparable a la respuesta Proporcional mostrada anteriormente, razón por la que se definieron cinco conjuntos tanto para la variable “setpoint” como para “floop”, ya que la primera contiene la configuración establecida por el usuario y la segunda es la realimentación de la salida al controlador difuso y cuya tendencia es la de igualar el valor de “setpoint”.

Puede verse en la figura 3.3 la representación gráfica de los conjuntos difusos correspondiente a la variable “setpoint”, la cual respeta la condición para generar un archivo de código ASM compilable establecida previamente, que son la de no incluir espacios en blanco para los términos con nombres compuestos de más de una palabra.

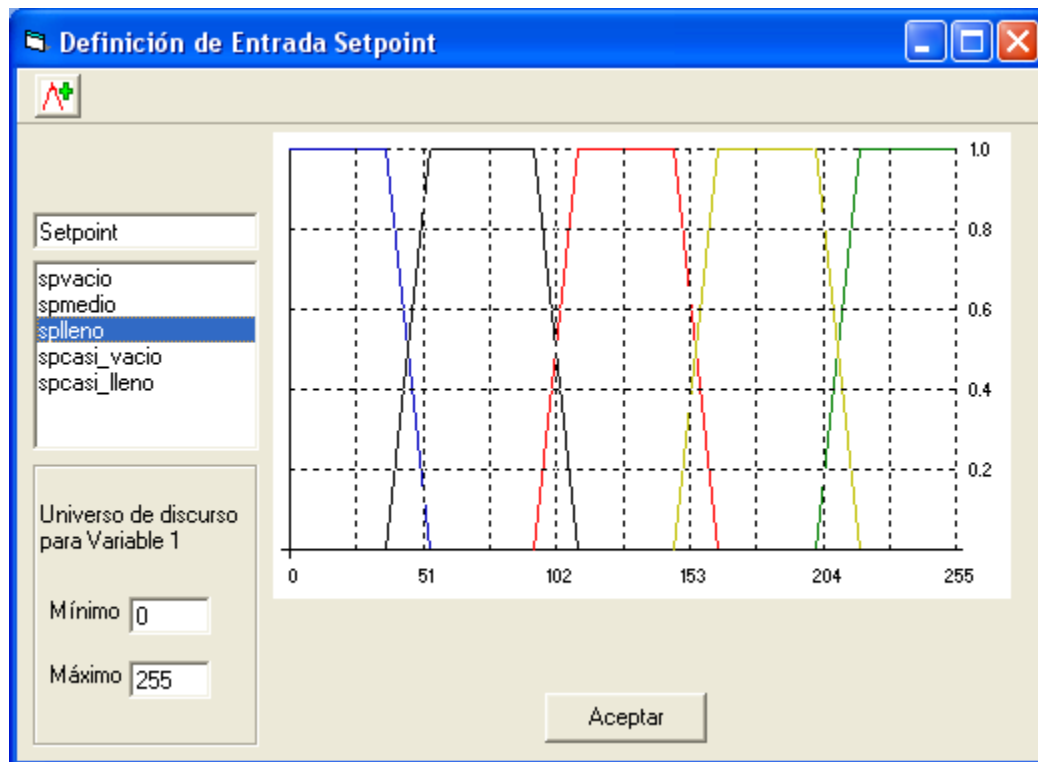


Figura 3.3. Representación gráfica de los conjuntos difusos para la variable “setpoint” del control de nivel de líquido.

Los nombres (o variables lingüísticas) asignadas a los conjuntos difusos definidos para la entrada setpoint son spvacío, spcasi_vacío, spmedio, spcasi_lleno, splleno. Como se observa en la figura 3.3, el universo de discurso varía de 0 a 255, se decidió usarlo así como una representación del volumen de agua contenido en el tanque, y a la vez para demostrar la versatilidad del controlador difuso, que no necesita exceso de cálculos y factores de conversión para realizar proporcionar una acción de control adecuada al sistema que rige, con lo cual bastará definir correctamente los conjuntos difusos sobre el

universo de discurso, así como las reglas de inferencia, para obtener una respuesta óptima del proceso.

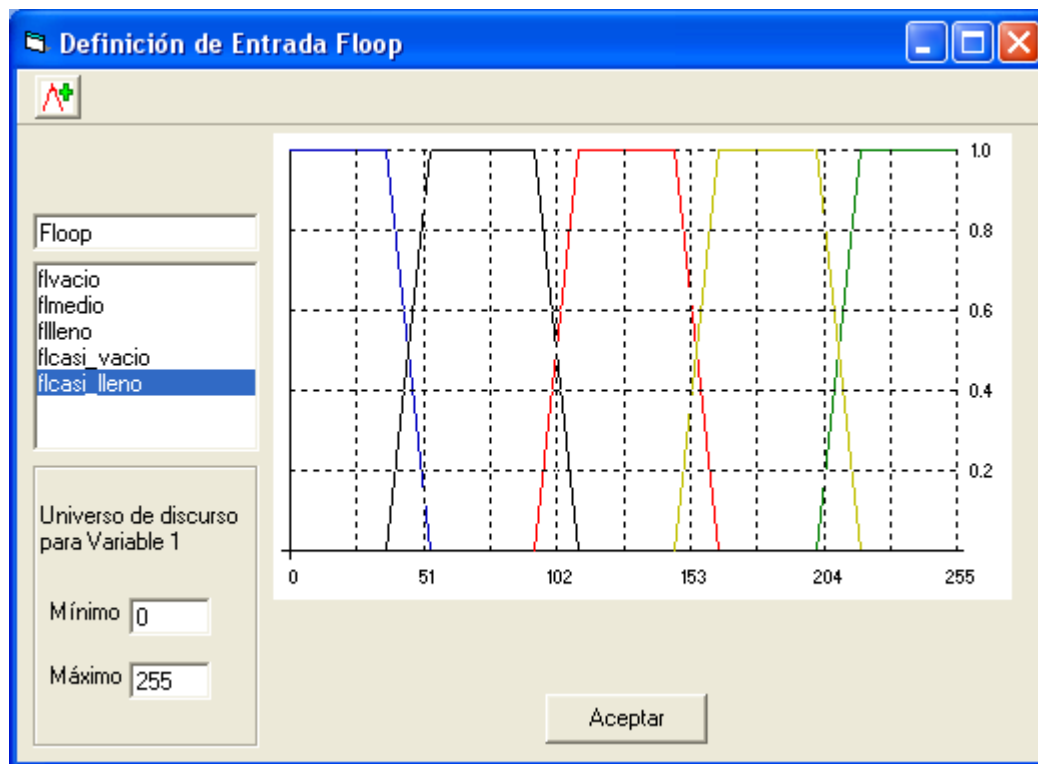


Figura 3.4. Representación gráfica de los conjuntos difusos de la entrada “Floop” del control de nivel de líquido.

La Figura 3.4 muestra la representación gráfica de los conjuntos difusos de la entrada Floop que corresponde a la realimentación de la salida (nivel de líquido medido en el tanque) al controlador difuso; como puede verse al igual que Setpoint respetan la condición de no incluir espacios en los nombres de conjuntos compuestos por más de una palabra, y, si se compara ambas figuras puede notarse que se han definido nombres similares, pero no repetidos que es la segunda condición a tener en cuenta a la hora de generar el código ASM compilable.

Para las salidas difusas del sistema se consideraron también cinco términos tipo barra, los cuales representarían el nivel de voltaje a suministrar a la bomba eléctrica, como puede

observarse en la gráfica de la figura 3.5 los niveles de activación para cada uno de los términos correspondían a los siguientes: Dec_alto 0%, Dec_medio 25%, Cero 50%, Inc_medio 75% e Inc_alto 100%. Ello se definió así en vista que el microcontrolador no es capaz de proveer por ningún puerto voltajes bipolares, por lo tanto las salida defuzzificada mediante la función WAV del microcontrolador MCU 68HC12 se aplicó al puerto B, el cual a su vez se conectó a una red R-2R para obtener de esta forma un voltaje proporcional al valor digital de salida, debido a que este voltaje tenía un discurso de 0 a 5 Voltios se le restó 2.5 Voltios fijo para obtener así un discurso de -2.5 a 2.5 Voltios, los cuales se aplicaron a un circuito el cual proporcionaba voltajes de aproximadamente 4 a 10 voltios para alimentara a la bomba eléctrica, ello se definió así considerando la “inercia” del sistema tanque-bomba eléctrica, ya que ésta última responde a magnitudes de voltaje pequeños pero no produce un caudal de entrada al tanque.

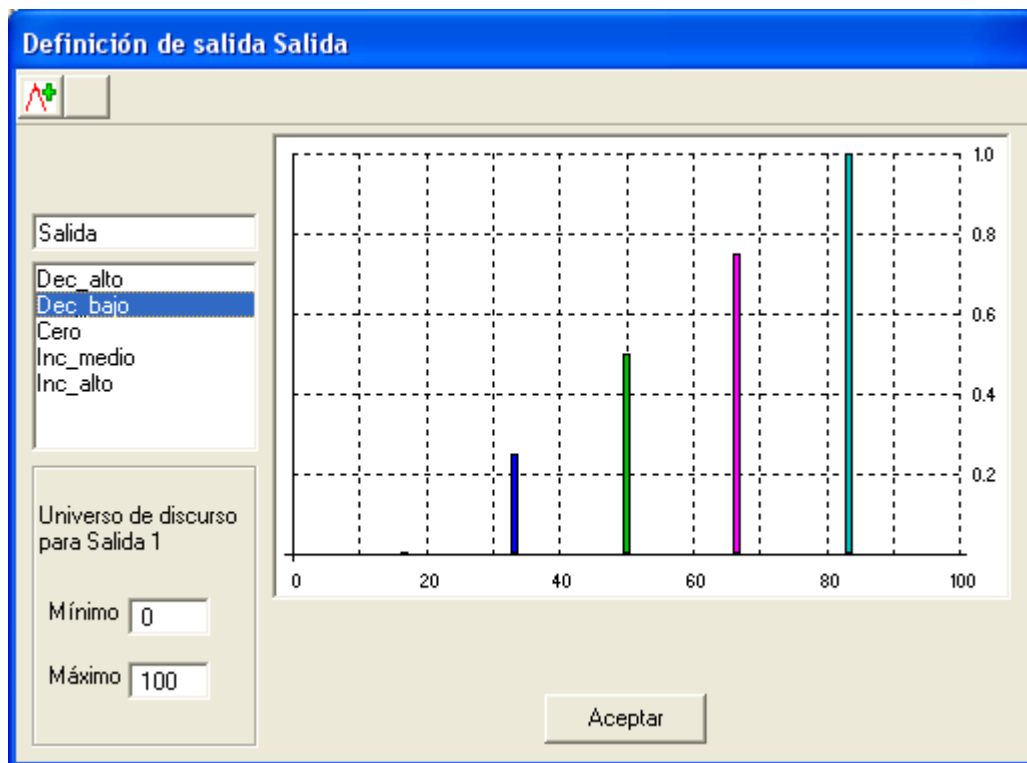


Figura 3.5. Representación gráfica de los términos de salida para el controlador difuso de nivel de líquido en un tanque

Edición del primer bloque de reglas

Cuando
 Setpoint
 sp_casilleno

floop
 fl_lleno

Entonces
 Salida
 Dec_medio

Aceptar

Setpoint

spvacio	Cero	Dec_alto	Dec_alto	Dec_med	Dec_alto
spmedio	Inc_alto	Cero	Dec_alto	Inc_medio	Dec_med
splleno	Inc_alto	Inc_alto	Cero	Inc_alto	Inc_medio
spcasi_vaci	Inc_medio	Dec_med	Dec_alto	Cero	Dec_alto
sp_casilleno	Inc_alto	Inc_medio	Dec_med	Inc_alto	Cero

fl_vacio fl_medio fl_lleno fl_casivacio fl_casilleno
 floop

Figura 3.6. Bloque de reglas definidas para el control difuso sobre un sistema de nivel de líquido.

La figura 3.6 muestra el bloque de reglas difusas definidas en este experimento, las reglas están mostradas en forma de matriz de dos dimensiones en donde las variables de entrada: Setpoint y floop definen dichas dimensiones, y los términos de las salidas están ubicados en los registros correspondientes a la intersección de los términos para las entradas. Cabe mencionar que la definición de las reglas difusas obedece a la asignación que de las salidas se hace por medio de una persona conocedora de la respuesta de la planta en términos operativos.

En la gráfica 3.6 es importante remarcar que, por características inherentes a la planta de nivel utilizada, la cual emplea una bomba motorizada para la inyección de líquido al tanque, dicha bomba debe vencer cierta inercia para iniciar el llenado del tanque, existen factores de uso de equipo, calibración del transductor de presión a voltaje, los cuales afectan la adversamente la respuesta del sistema ante los valores de voltaje de entrada, lo cual representó un obstáculo en la implementación del controlador difuso, para lo que se hizo necesario aplicar un nivel de voltaje de offset, el cual se determinó de la observación luego de la experimentación con el controlador y la respuesta del sistema ante este para los diferentes valores de setpoint y los conjuntos difusos definidos; razón por la cual se sacrificó exactitud en la respuesta a los conjuntos difusos Vacío y Lleno como podrá observarse.

Al efectuar el experimento se obtuvieron los resultados que se presentan a continuación

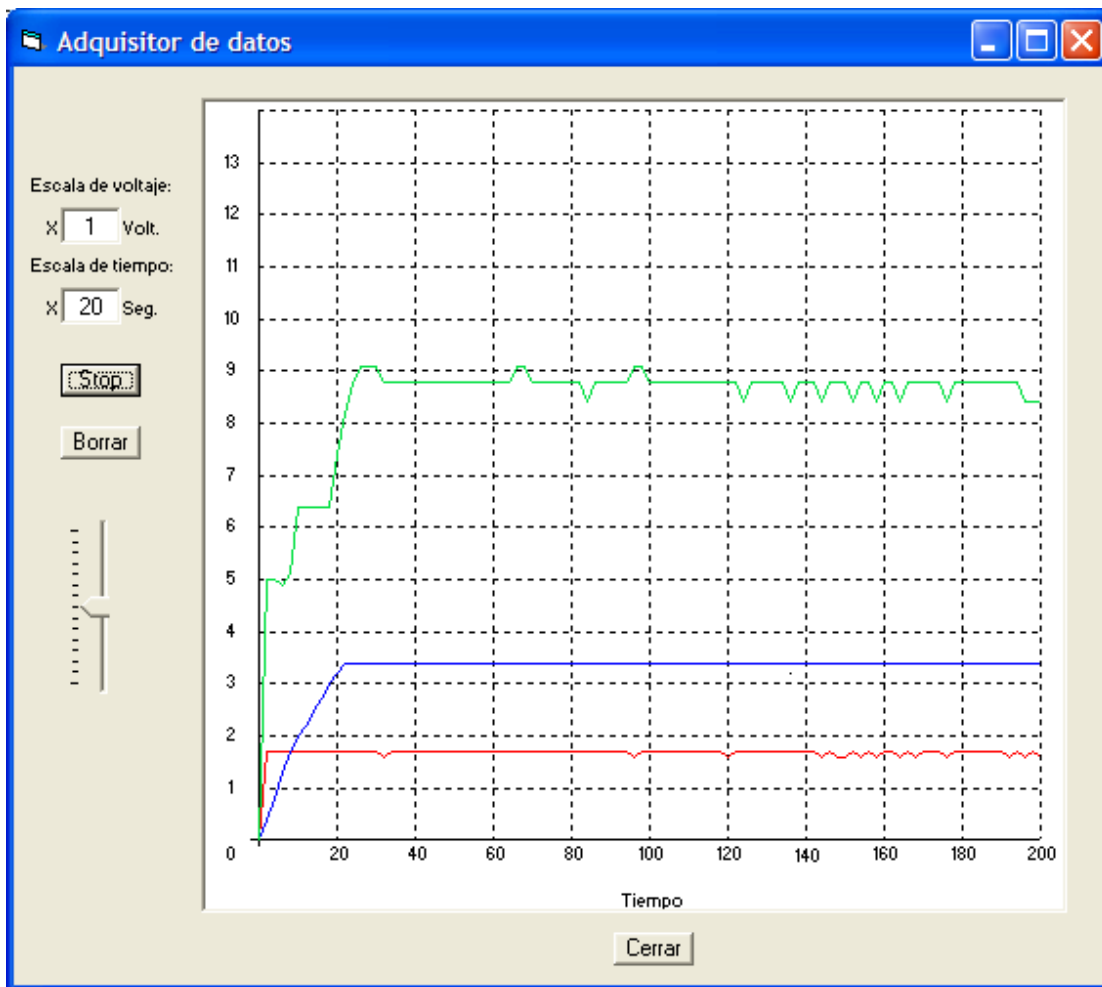


Figura 3.7. Representación gráfica de un controlador difuso para nivel casi vacío.

En la figura 3.7 se observan tres representaciones: La gráfica de color verde representa el valor de la acción de control suministrada al controlador, dicho valor es superior en magnitud al valor de salida del controlador ya que el controlador a su salida tiene definidos términos en la salida, las cuales estaban definidas como incremento bajo, incremento medio, cero, incremento alto; dicho voltajes eran representativos de la salida teniendo en cuenta que la salida del MCU se obtenía una señal de 0 a 5 V y posteriormente por hardware se adecuaba a un nivel de voltaje. La gráfica de color azul es la salida del sistema controlador proveniente del transductor de presión, el cual se utilizó para obtener una representación en voltaje del nivel en el tanque, dicha magnitud de voltaje fue necesario adecuarla para poder suministrarla a la entrada del MCU, el cual

posee un Convertidor de Análogo a Digital con un rango de voltajes de 0 a 5 Vdc. De la misma gráfica azul puede observarse el tiempo que le tomó al controlador difuso llegar a la estabilidad para el valor de referencia, el cual fue de poco más de veinte segundos. La gráfica de color rojo es el valor de referencia definido por el usuario para el controlador, para el caso de la representación gráfica de la figura 3.7 el valor de referencia se encuentra dentro del conjunto definido lingüísticamente como casi vacío.

Cabe mencionar que la respuesta del controlador difuso fue registrada por medio de la aplicación del Adquisidor de Señales, obteniéndose por el puerto serie los datos de respuesta que el MCU suministro a la planta.

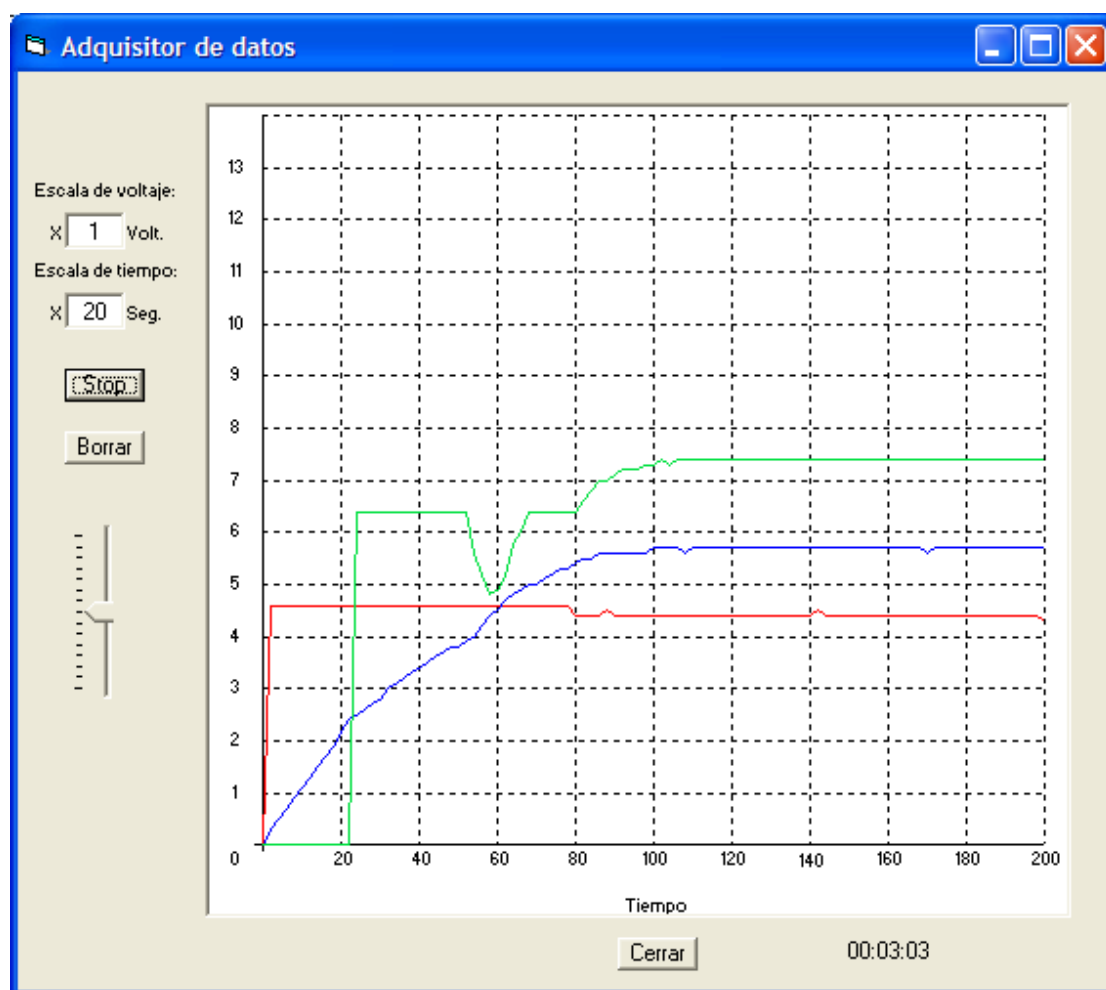


Figura 3.7. Representación gráfica de un controlador difuso para nivel medio

En la figura 3.7 se presenta la respuesta del controlador ante un valor esperado de nivel medio, siendo siempre la salida la figura representada con el color azul. En esta gráfica puede observarse que el tiempo de estabilización es de alrededor de noventa segundos y que el valor de sobre impulso en la respuesta es prácticamente nulo.

Para un valor de referencia del 75% se obtuvo la gráfica de la figura 3.8, en este se compara prácticamente el controlador difuso con el controlador proporcional presentado anteriormente, puede observarse que el tiempo de estabilización en el valor de referencia es prácticamente igual a cincuenta segundos tanto usando el controlador difuso como el controlador proporcional convencional.

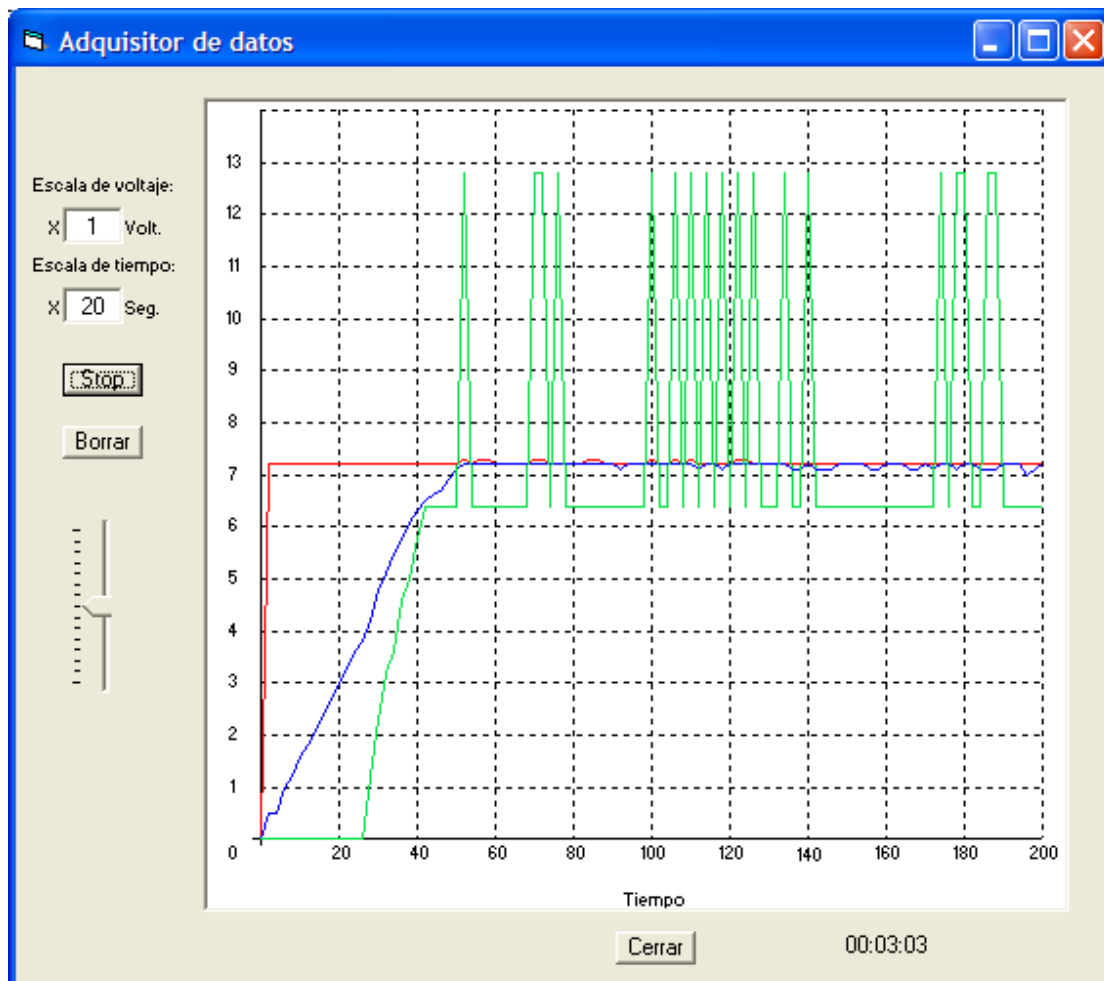


Figura 3.8. Representación gráfica de un controlador difuso para nivel casi lleno

En la figura 3.9 se presenta el resultado obtenido para un valor de referencia dentro del conjunto difuso lleno, ante el cual el comportamiento del controlador arrojó como resultados (a) un tiempo de estabilidad de alrededor de 25 segundos y (b) un sobre impulso prácticamente nulo; los valores de escala para la graficación fueron de 1V y 10 segundos.

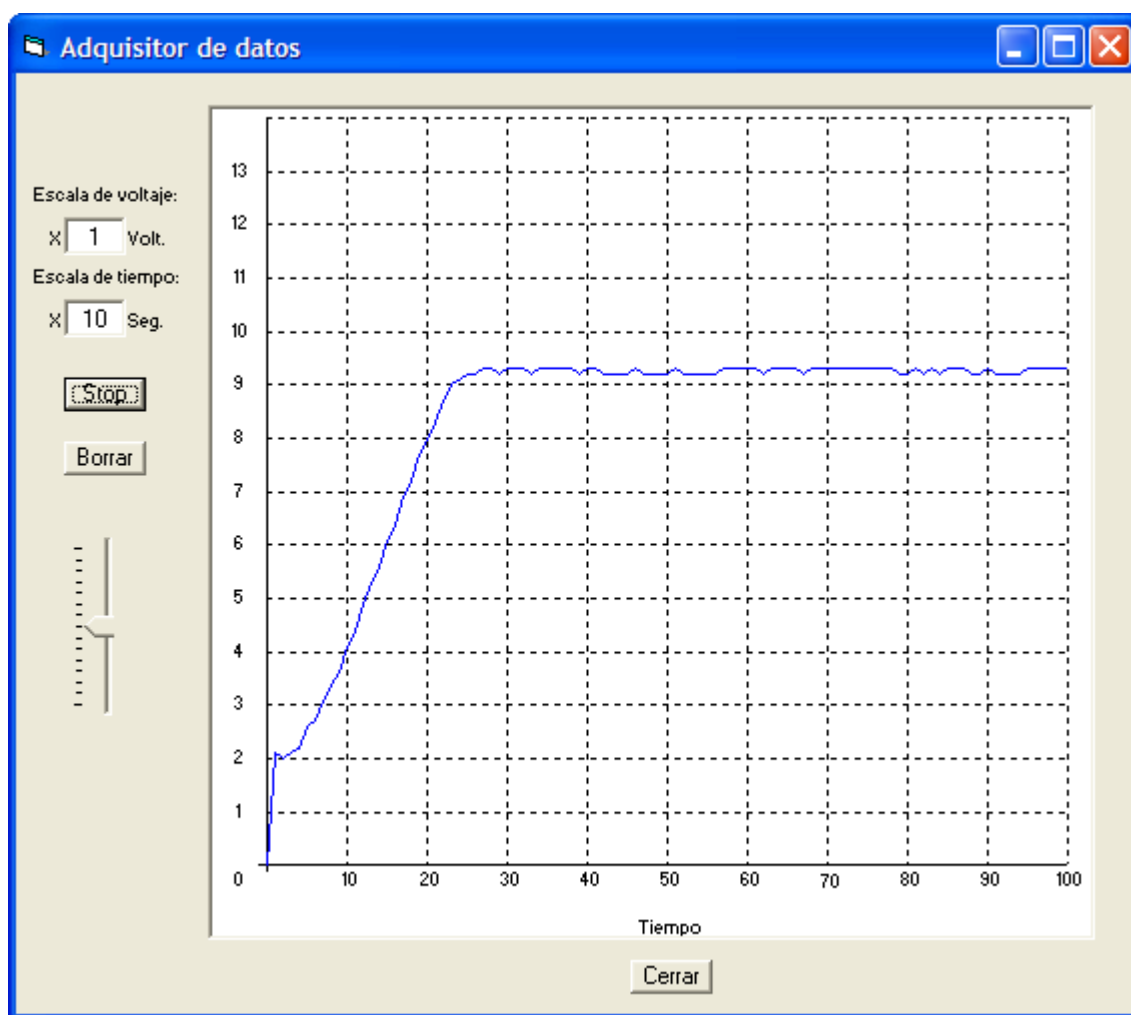


Figura 3.9. Respuesta obtenido para setpoint lleno con un controlador difuso en el sistema de nivel de líquido

3.2 SISTEMA DE CONTROL DE VELOCIDAD

El sistema de control de velocidad se realizó con un motor de DC perteneciente al laboratorio de Instrumentación y Control del Centro de Investigaciones y Transferencia de Tecnología de la Universidad Don Bosco, para la medición de la respuesta se utilizó el encoder instalado en dicho motor y el transductor de velocidad (n) a voltaje (V) Lucas Nulle.

Dentro de los aspectos a considerar para la realización de un control de velocidad esta el factor de escala para el tiempo, ya que la respuesta del controlador debe ser adquirida para su análisis en un tiempo pequeño comparado con el sistema de control de nivel en un tanque como el presentado en el experimento anterior.

El control automático de velocidad para un motor de DC se basa en el principio que en dicho motor la velocidad angular a la que gira su eje es directamente proporcional al voltaje suministrado al motor, por lo que para aumentar la velocidad de giro basta con aumentar la magnitud del voltaje suministrado y para reducir la velocidad se suministra un voltaje menor al motor.

En la gráfica 3.10 se muestra la respuesta en lazo abierto para el motor de DC, para la cual se abrió el lazo de realimentación proveniente del transductor de velocidad a voltaje.

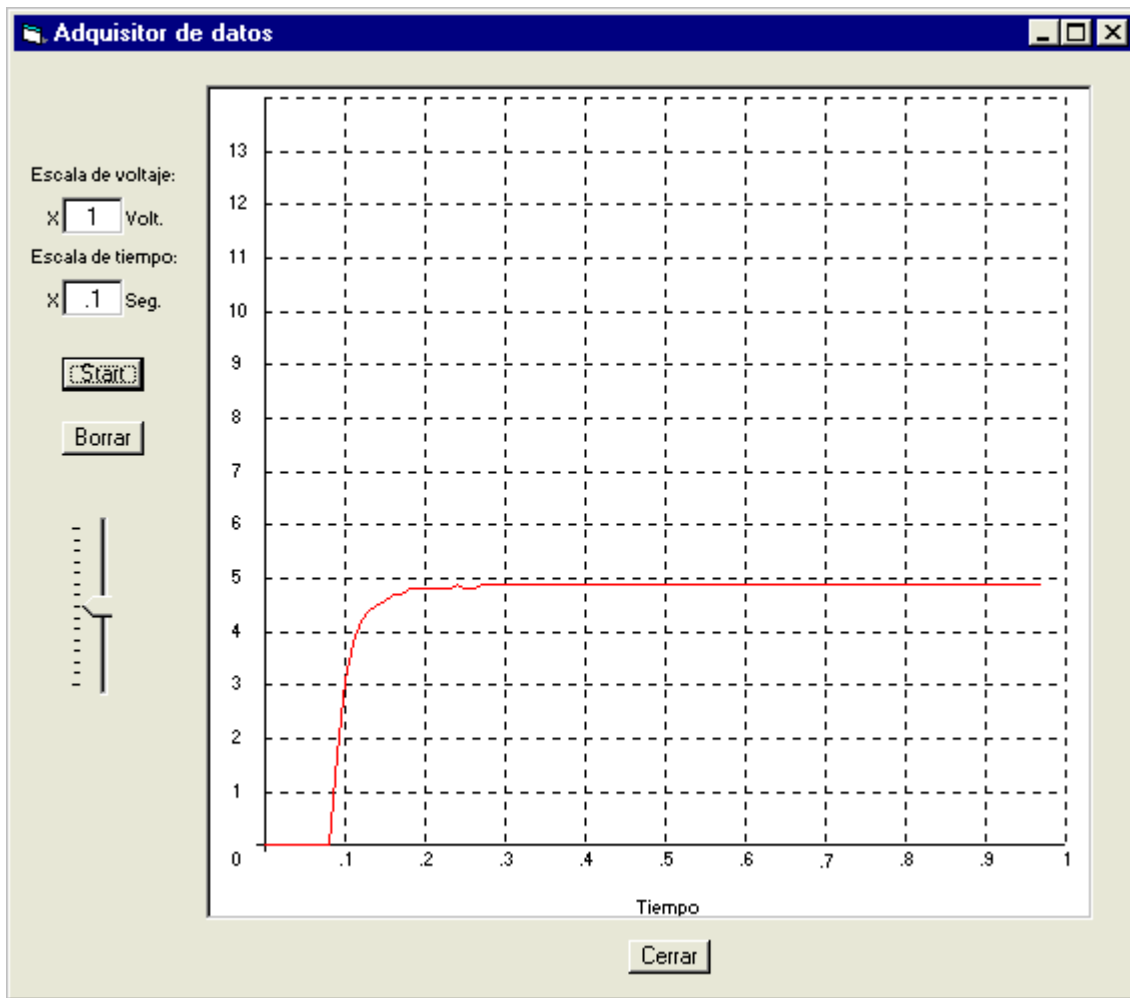


Figura3.10. Gráfica de respuesta para lazo abierto para el motor de DC

En la figura 3.10 se presenta la respuesta para un valor de referencia de 50% a lazo abierto utilizando el motor de DC del CITT, en dicha gráfica se observa que le tomó alrededor de 0.5 segundos el llegar al valor de referencia.

Utilizando el controlador Proporcional del CITT a lazo cerrado y con un valor de referencia de 50% del valor máximo se obtuvo la gráfica mostrada en la figura 3.11.

De la figura 3.11 puede verse que el tiempo que le tomó al controlador llegar a un punto de estabilidad para el valor seteado fue de aproximadamente medio segundo y que el valor del sobreimpulso fue prácticamente nulo.

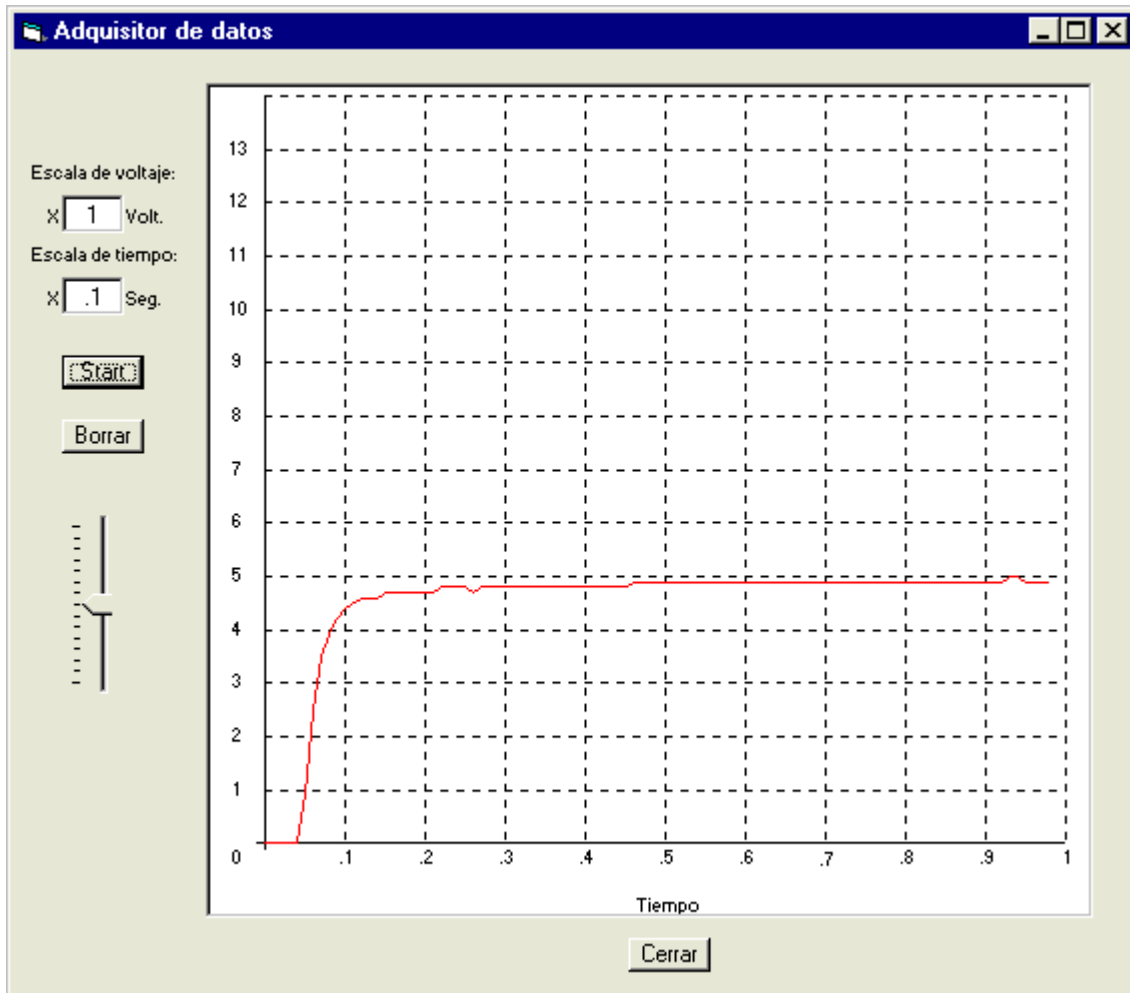


Figura 3.11. Gráfica de respuesta de la velocidad de un motor eléctrico bajo la acción de un controlador proporcional

Para la realización del experimento con controlador difuso se tomaron en cuenta los conjuntos difusos de entrada representados en la figura 3.12:

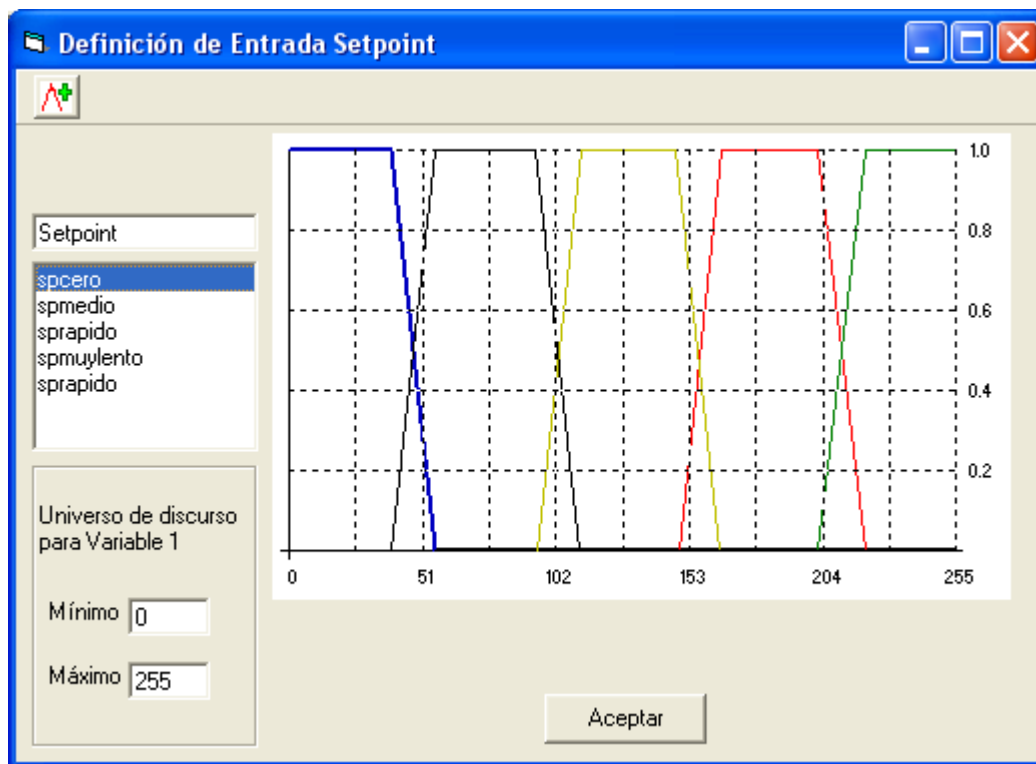


Figura 3.12. Representación gráfica de los conjuntos difusos para el control de velocidad de motor DC (setpoint)

Los conjuntos definidos son cinco para la entrada difusa Setpoint, dichos conjuntos son los términos definidos para el sistema de control difuso implementado en la planta de control de velocidad de un motor DC, las definiciones lingüísticas para estos términos son las siguientes: spcero, spmuylento, splento, spmedio y sprápido, correspondiendo estas a distintos valores en velocidad angular para el motor DC.

Se utilizó una entrada correspondiente al lazo de realimentación denominado floop, los conjuntos de entrada difusos definidos para dicha entrada son:

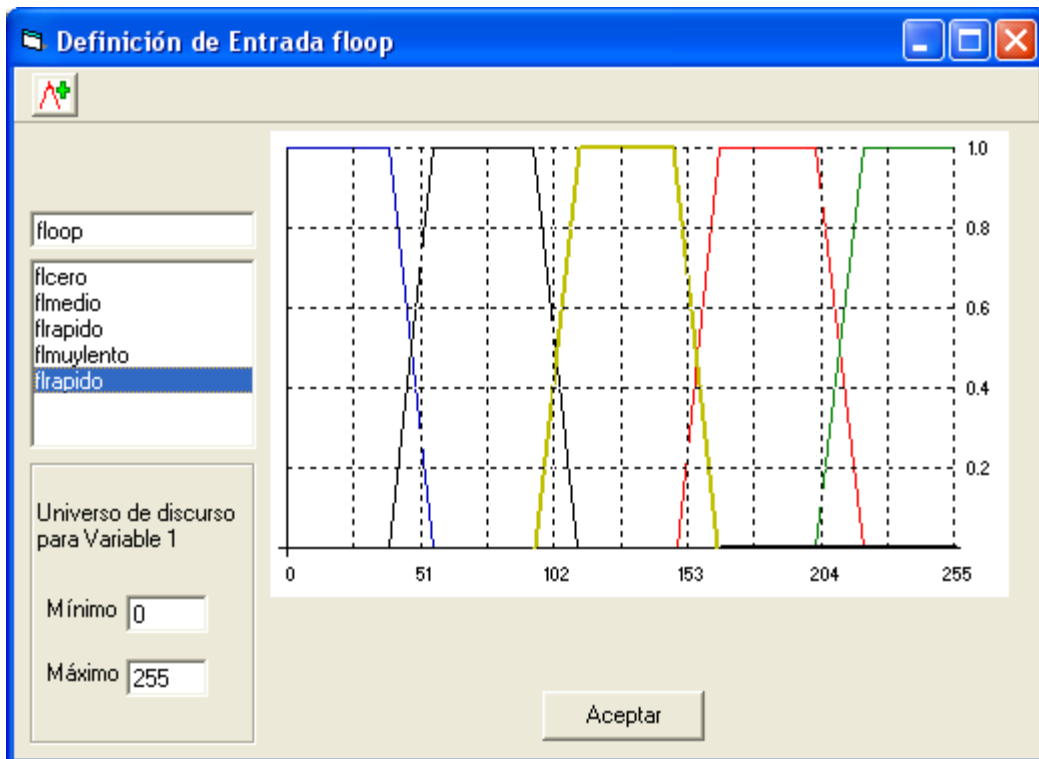


Figura 3.13. Representación gráfica de los conjuntos difusos para el control de velocidad de motor DC (floop)

En la figura 3.13 se muestran los conjuntos de entrada difusos definidos para la entrada floop , en este caso se decidió la implementación con cinco términos para la entrada floop, tomándose a estos también como trapezoidales en un universo de discurso de 255; siendo estos términos flcero, flmedio, flrapido, flmuylento, flrapido.

Los términos definidos para la salida del sistema se muestran en la figura 3.14: nocambio, incmedio, incalto, decmedio y decalto. Los valores de activación determinados para cada uno de estos términos son de 0, 25, 50, 75 y 100% respectivamente.

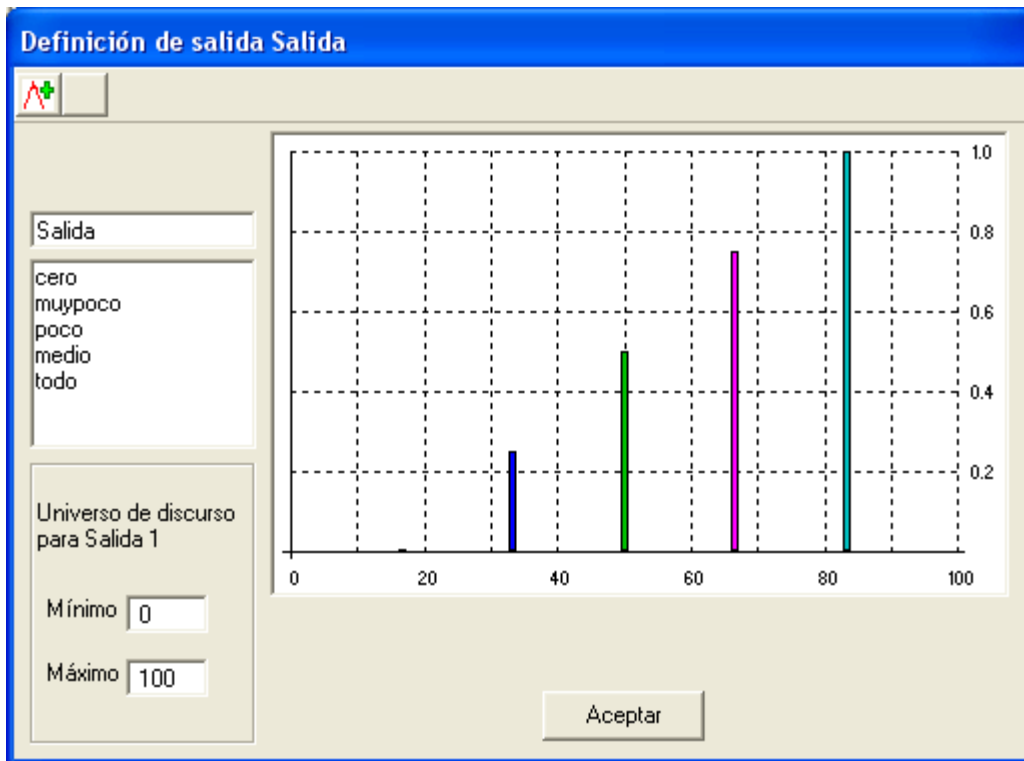


Figura 3.14. Definición de términos de salida

Una vez definidos los conjuntos correspondientes a las diferentes entradas y los términos tipo singleton para la salida, puede continuarse con la generación de las reglas de inferencia difusa en la ventana de Edición del bloque de reglas, en la cual se cuenta con una barra de herramientas y cajas de texto desplegables con los nombres de las variables utilizadas y a su vez, dentro de estas listas se presentan los términos de salida, para el caso presentado se contaba con dos variables de entrada con cinco términos cada una, si existiesen más variables habrá tantas cajas de texto como variables de entrada hasta un límite de cinco. Puede verse que los antecedentes están encerrados en un recuadro denominado “cuando” y los consecuentes se encuentran en un recuadro denominado “entonces”, lo cual hace alusión a las reglas SI ENTONCES utilizadas en la lógica difusa para realizar la acción de control; Así mismo se presenta una matriz de texto sobre la cual se establecerá la relación entre antecedentes y consecuentes de forma gráfica, la misma

tendrá tantas filas como términos tengan la variables Setpoint para este caso particular y tantas columnas como términos tengan la variable Floop.

Edición del primer bloque de reglas

Cuando
 Setpoint
 spmedio

floop
 flento

Entonces
 Salida
 medio

Setpoint
 spcero
 spmedio
 sprapido
 sprmuy lento
 splento

cero	cero	cero	cero	cero
medio	medio	cero	medio	medio
todo	todo	todo	todo	todo
muy poco	cero	cero	muy poco	cero
poco	cero	cero	poco	poco

flocero flmedio flrapido flmuy lento flento
 floop

Aceptar

Figura 3.15 Representación gráfica de las reglas difusas definidas para el control de velocidad.

En la figura 3.15 se muestra la matriz de reglas difusas correspondientes al control de velocidad de un motor de DC. En la representación matricial pueden observarse las relaciones existentes entre los términos definidos para las entradas y las salidas esperadas para cada uno de las reglas que el ente experto ha determinado convenientes.

Será con estas reglas que la aplicación generará el código en un archivo ASM compilable para el MCU 68HC12.

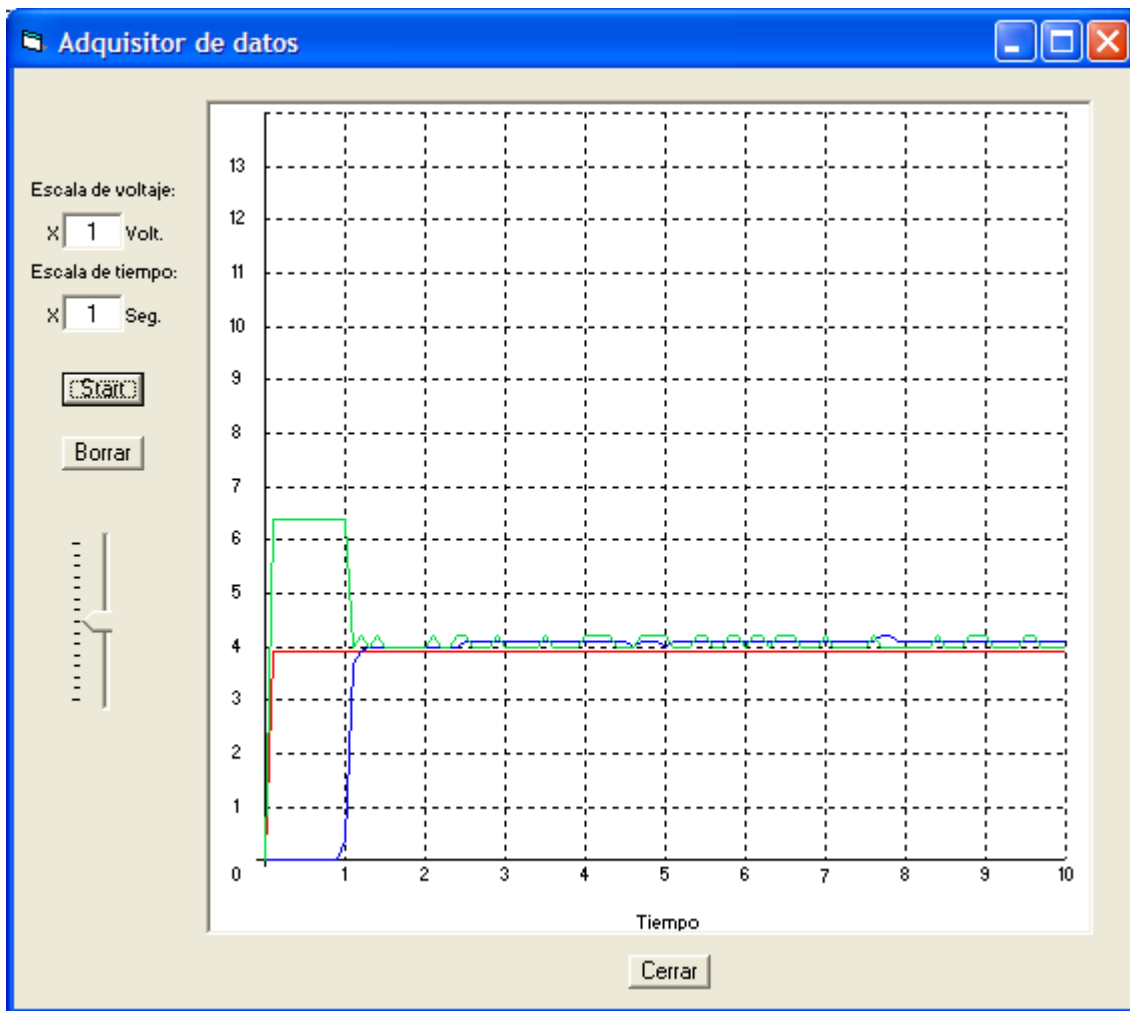


Figura 3.16. Gráfica de respuesta de la velocidad de un motor eléctrico bajo la acción de control de un controlador difuso.

En la gráfica 3.16 se presenta la acción de control sobre el motor en color verde, el punto de referencia se muestra en rojo y la respuesta del controlador en color azul.

Puede observarse en la gráfica verde, la acción de control, el comportamiento de ésta cuando se arranca el sistema ya que entran en juego las reglas definidas por el usuario, en este caso el valor de la acción de control proporciona al motor voltaje alto para llegar al nivel de referencia pero es hasta que este empieza a responder que la acción de control

inicia su función de recalcular la corrección del valor actual para llegar al punto de referencia.

El tiempo que le tomó al controlador difuso en llegar a un valor de estabilidad fue de un segundo. Realizando un acercamiento a la gráfica puede verse con más detalle la acción de control para el sistema de control de velocidad con un valor de referencia de alrededor del 50% del valor máximo, tal como se muestra en la figura 3.16

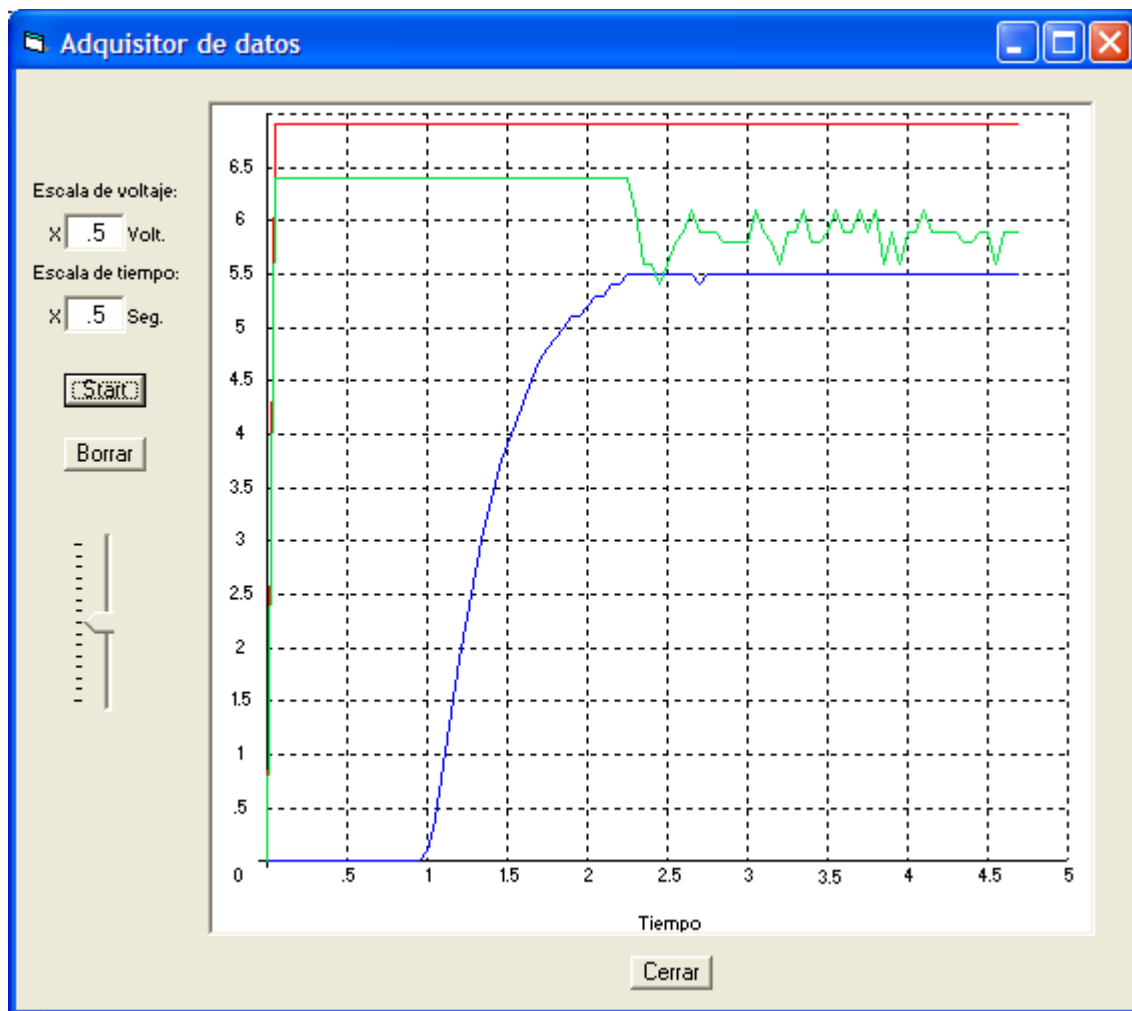


Figura 3.17. Acercamiento de la respuesta de la velocidad de un motor eléctrico bajo la acción de control de un controlador difuso con un valor de referencia de 50% del valor máximo.

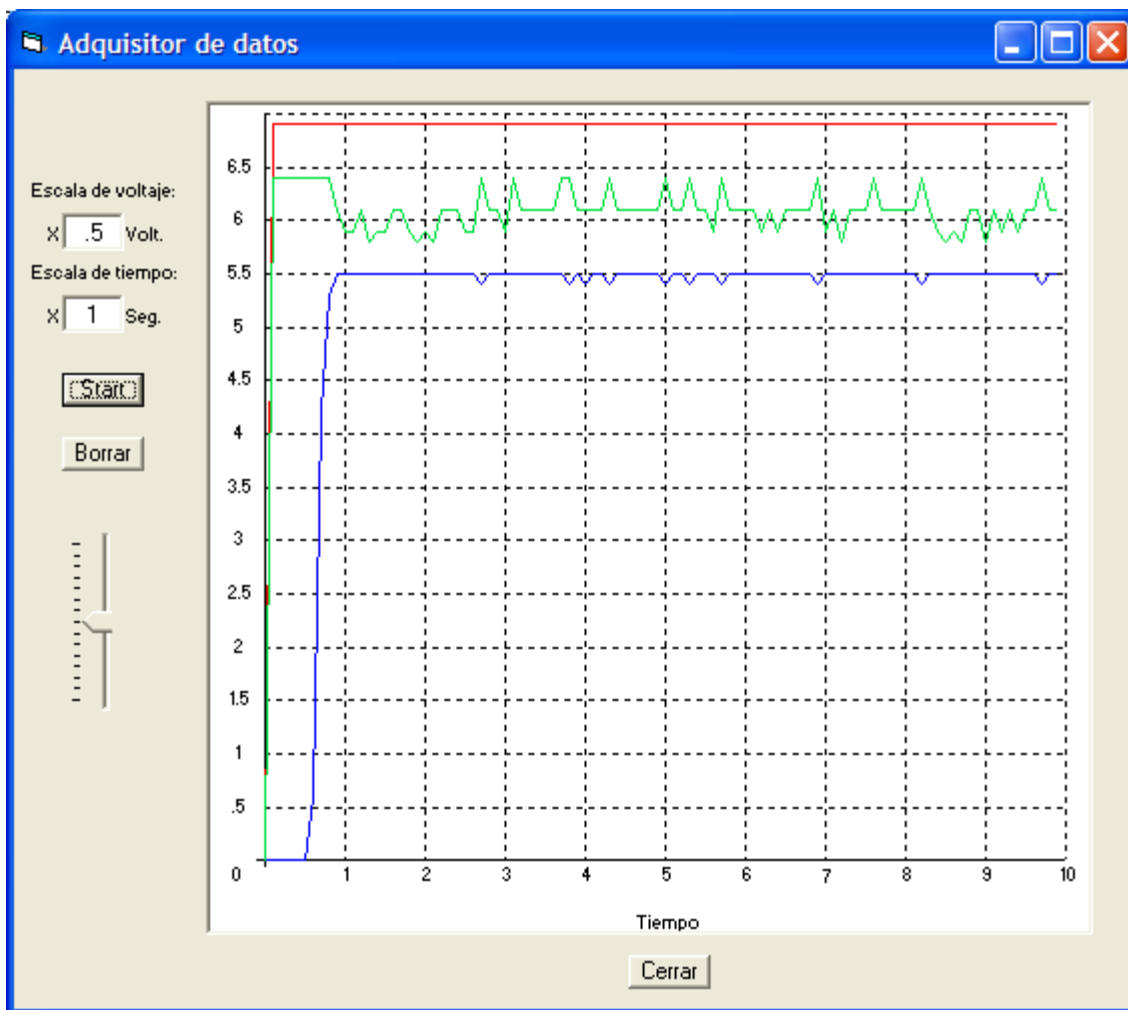


Figura 3.18. Respuesta del controlador difuso ante perturbaciones

En la figura 3.18, se presenta una gráfica en donde ante un valor de referencia de 50% del valor máximo, se sometió al sistema a una serie de perturbaciones, las cuales provocaban una variación en la acción de control, la cual reponía de acuerdo al cambio en la salida del sistema cuando se sometía éste a una perturbación.

3.3 RESULTADOS DEL CONTROL DIFUSO EN CENTRIFUGADORA DE ACEITE

El control actual en la centrifugadora se instaló originalmente en la Planta Nejapa Power¹⁰ dejando seteos que hasta la fecha no han sido auditados, teniendo como parámetro el funcionamiento aceptable que para el proceso la respuesta del controlador brinda; por medio de la utilización de la Aplicación del Adquisidor de Datos se obtuvo una gráfica de la respuesta en el tiempo como se muestra en la figura 3.19

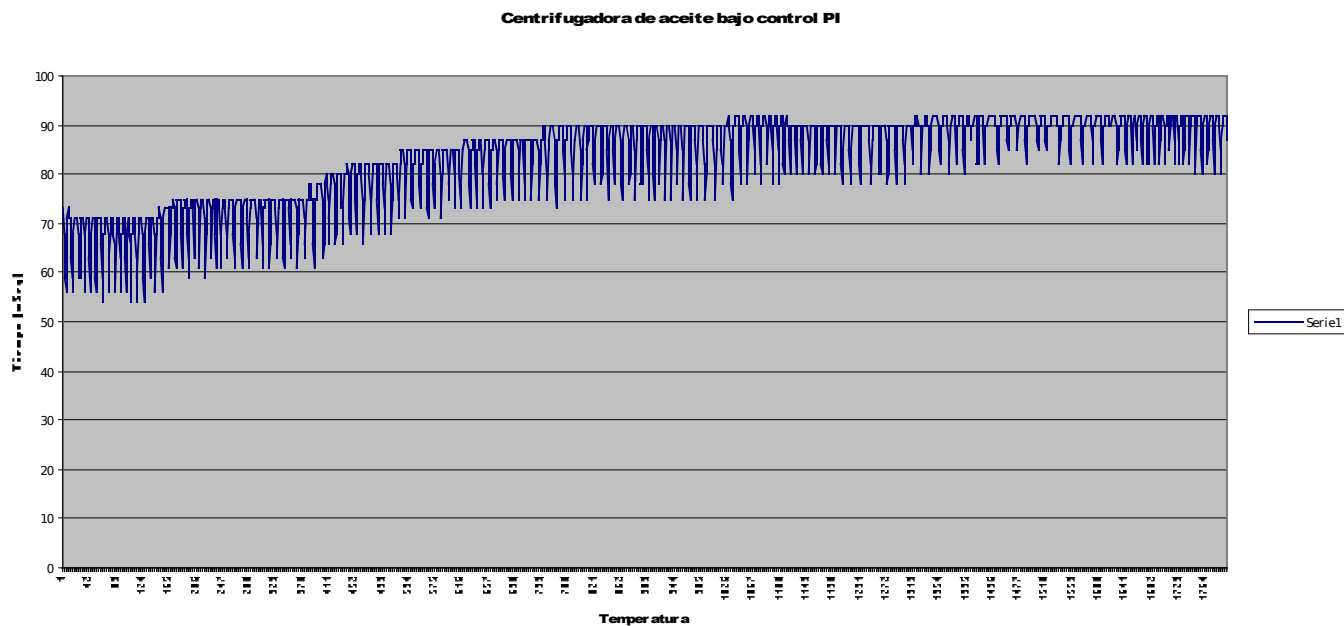


Figura 3.19. Respuesta del controlador instalado

¹⁰ Planta Generadora de Energía Eléctrica perteneciente a El Paso Technolgy El Salvador.

A continuación se presenta un acercamiento del controlador actual:

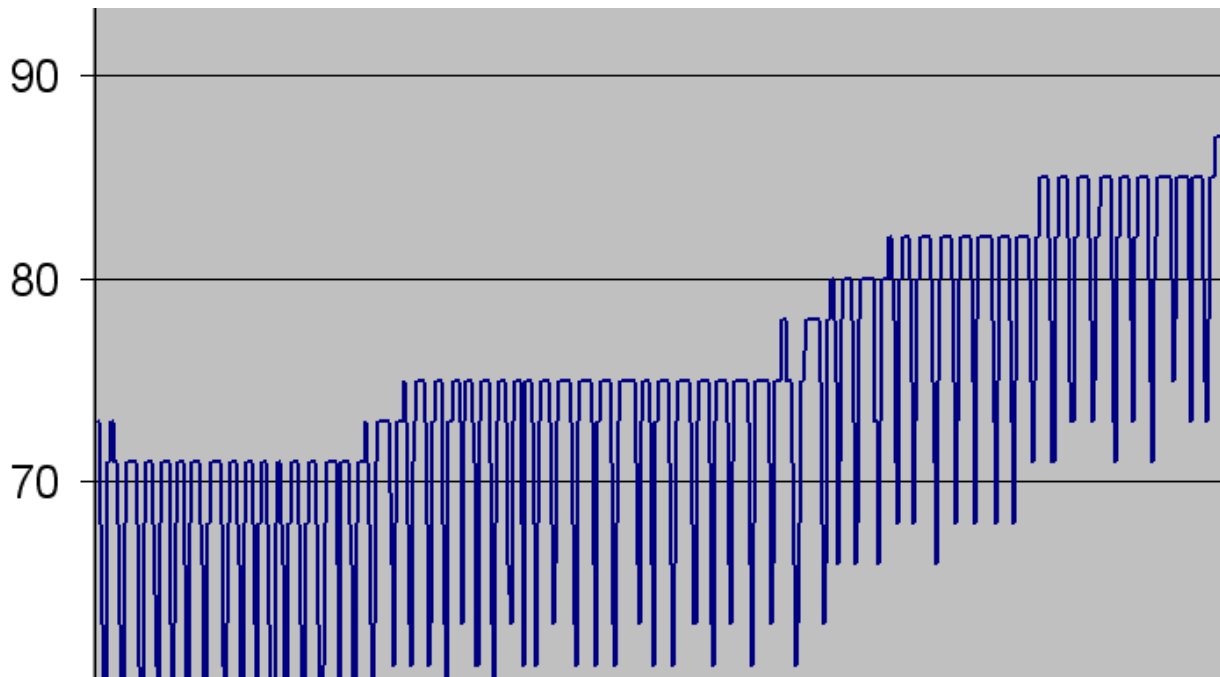


Figura 3.20. Acercamiento de acción de control actual

Como puede verse en la figura 3.20 las variaciones en la respuesta del controlador se deben a un ajuste inadecuado de los parámetros de control PI pero que debido a la naturaleza del proceso y a que no se cuenta con un registro de una gráfica en el tiempo de la respuesta del controlador se ha tomado dicha respuesta como aceptable. El tiempo que llevo al controlador llegar al valor programado fue de seiscientos sesenta segundos equivalentes a once minutos.

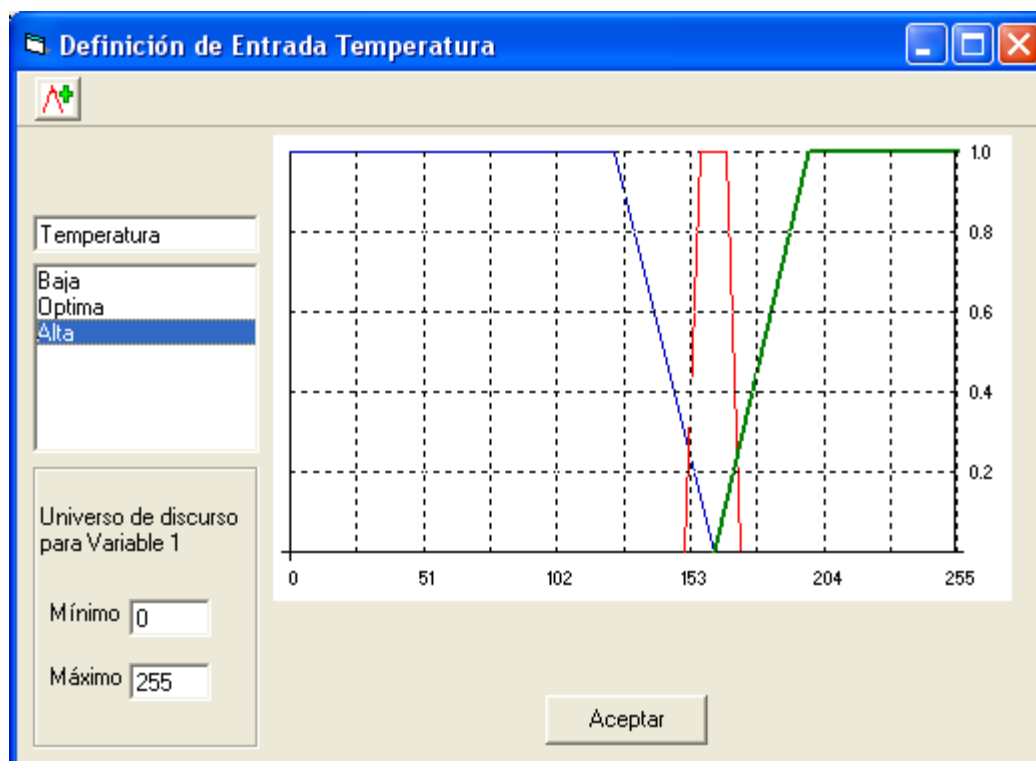


Figura 3.21. Representación gráfica de los conjuntos de entrada para el control de temperatura en la centrifugadora

En la figura 3.21 se presentan los tres términos difusos definidos para la realización del control de temperatura en una centrifugadora de aceite lubricante, los conjuntos definidos son: baja, optima y alta.

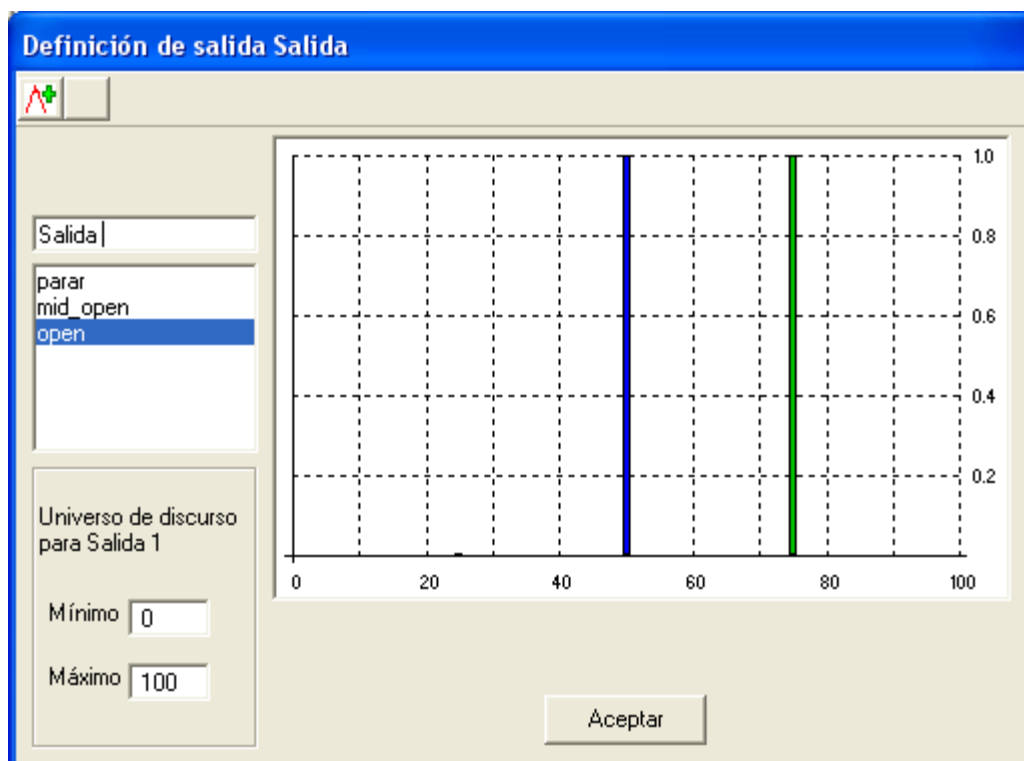


Figura 3.22. Representación gráfica de los términos de salida para el control de temperatura

En la figura 3.22 se muestran los términos definidos para las salidas así como su grado de activación, para este caso se definió para con un 0%, abrir_medio y abrir_todo con un 100% de grado de activación.

Edición del primer bloque de reglas

Quando
Temperatura
Alta

Entonces
Salida
mid_open

Aceptar

Temperatura	
Baja	open
Optima	parar
Alta	mid_open

Figura 3.23. Representación gráfica de las reglas difusas definidas para el controlador de temperatura.

En la figura 3.23 se presenta la matriz de reglas difusas para el controlador de temperatura en la centrifugadora de aceite lubricante, definiéndose para ello una entrada proveniente de la señal del PT-100 y la salida referente a un actuador consistente en una válvula motorizada.

Como es de esperar un controlador difuso solamente es capaz de entregar una salida digital y esta a su vez puede ser pasada a un nivel de voltaje, cuya magnitud sea representativa del valor digital presentado, más para este caso se contaba con la válvula motorizada antes mencionada cuyo funcionamiento es según el siguiente detalle:

Voltaje de entrada: 24Vac

Total de vueltas para carrera completa del émbolo estrangulador: 23

Tiempo total de carrera completa: 120 segundos

Por lo tanto se hizo necesario el adecuar la salida digital a las características de la válvula, se pensó en activar el motor y verificar el desplazamiento del émbolo para que la apertura fuese de cierta forma proporcional a la acción de control, más por diversos inconvenientes, además de los mecánicos a la hora del reemplazo de la misma se decidió no usarla; la siguiente idea fue la de fabricar un codificador que pudiera conectarse al eje del motor y así medir el desplazamiento angular del mismo, como otra forma de hacer que la acción de control fuera proporcional a la apertura de la válvula más se presentó el inconveniente de la vibración producida por el mecanismo de la misma que ocasionaba que el encoder se desconectara del eje del motor ocasionando medidas erróneas por difertes razones como el deslizamiento mecánico, obstrucción de los dispositivos ópticos y ruido magnético, por lo que se descartó la opción (cabe aclarar que las opciones antes mencionadas no eran del todo inútiles sino más bien sobre la base de los recursos con que se contaba, experimentación y condiciones de trabajo se decidió no implementarlas).

Por las razones antes mencionadas se decidió el hacer que la acción de control pudiera controlar la válvula controladora del flujo de vapor a la centrifugadora mediante una base de tiempo de activación de la misma y cuyo giro estuviese determinado por la magnitud de la acción de control, efectuando el control de la forma como se describe a continuación:

Desplazamiento angular total = θ

Total de vueltas= rev

Tiempo de carrera completa= τ

Paso angular= ϕ

Base de tiempo= t_s

$$\theta = 360 \times rev = 360 \times 23 = 8280^\circ$$

$$\phi = \frac{\theta}{255} = \frac{8280}{255} = 32.47^\circ$$

$$t_s = \frac{\tau}{\theta} \times \phi = \frac{120seg}{8280^\circ} \times 32.47^\circ = 470.58mseg$$

Sobre la base de los cálculos anteriores, se puede observar que el ángulo barrido por los engranajes del motor de la válvula actuadora, para una carrera completa del émbolo estrangulador es de 8280° que representa a 23 revoluciones, entonces si se sabe que la acción de control es capaz de variar entre 0 y 255 por lo cual el paso que pudiera representar el cambiar en uno el valor de la salida implicaría el girar 32.47° , y el tiempo que necesitaría estar activado el motor de la válvula para realizar ese movimiento es la relación descrita para t_s que involucra el ángulo de barrido para una carrera completa dividido por el tiempo que le toma al sistema recorrer dicho desplazamiento angular multiplicado por el paso angular que es capaz de desplazarse el sistema, que nos da una base de tiempo de 470.58mseg, dicho tiempo puede generarse mediante el sistema del Timer Output Compare del MCU68HC12 en conjunto con el sistema Acumulador de Eventos, cuya interrelación es la siguiente: mediante el Timer Output Compare, se genera una señal cuadrada que estará habilitada una vez que el MCU haya calculado la acción de control correspondiente a las condiciones actuales de entrada; el período de dicha señal debe ser igual a $t_s=470.58mseg$, lo que implica que debe poseer un duty cycle de la mitad de dicho tiempo, esta salida a su vez se debe realimentar al pin PT7 del microcontrolador que es el correspondiente a la señal PAI para el sistema contador de eventos, el cual debe

configurarse como interruptible para que el MCU de servicio a dicho subsistema cada vez que ocurra una transición positiva en la entrada antes mencionada, lo que en código de programación debe comparar la ocurrencia de dicho evento con la acción de control calculada por el microcontrolador, lo cual implica que se tiene un timer cuyo tiempo de activación corresponde a la acción de control proporcionada por el controlador difuso, y que a su vez se usa para controlar el motor de la válvula actuadora mediante un circuito acondicionador de señal descrito más adelante.

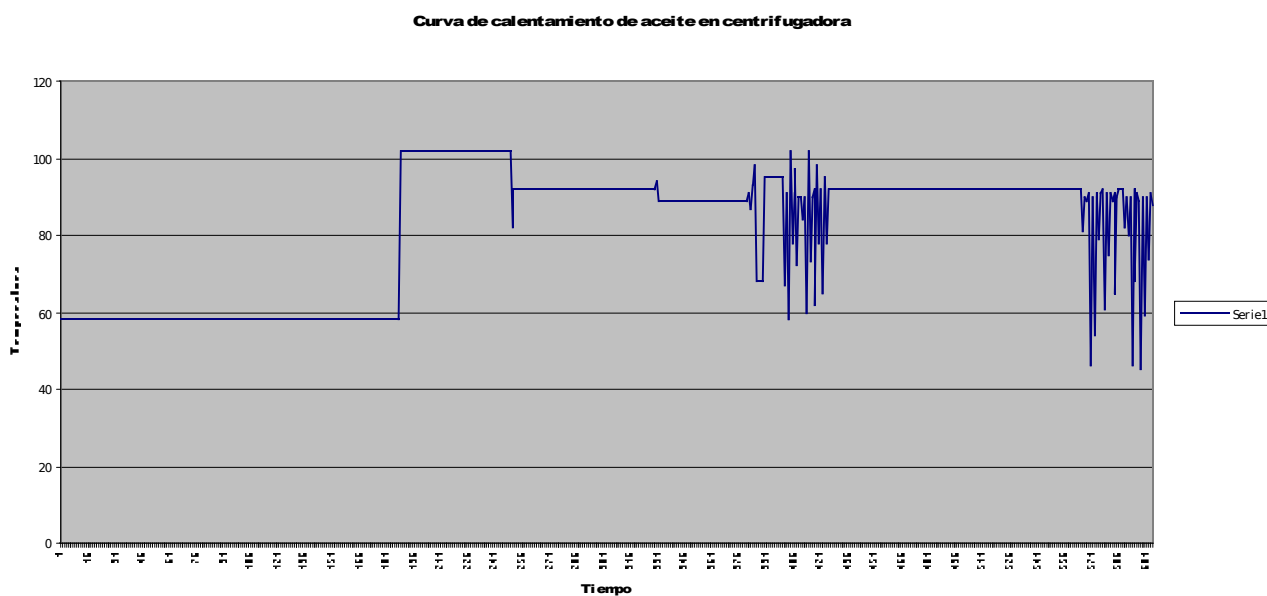


Figura 3.24. Curva de respuesta de Controlador Difuso implementado con MCU 68HC12

Según se observa en la figura 3.24 se obtuvo una mejor respuesta en el tiempo con el controlador difuso, llegando a la estabilidad en doscientos trece segundos equivalentes a tres minutos con treinta y tres segundos. Como puede observarse el hecho de llegar más rápido al valor establecido implica un sobreimpulso de 10 grados sobre el valor nominal.

Es de tomar en cuenta que a la hora de realizar las pruebas con el prototipo de Controlador Difuso en la Centrífugadora se observó el efecto de variables externas no

relacionadas directamente con el proceso y que en el caso de dejar permanentemente el controlador en operación deben mejorarse como lo es la protección contra interferencias, ya que se desarrollo la prueba bajo un ambiente de alta interferencia, ruido, vibración, arranque de motores de inducción, arranque de motores de combustión interna, factores que sumandos influyen en la respuesta que el controlador brinda ante las condiciones de sus entradas. Para mejorar el desempeño se recomienda el uso de gabinetes adecuados, preferiblemente IP65 debido a la ubicación, y con sistemas de apantallamiento y blindaje de cables en los transductores y actuadores adecuado.

Dada la naturaleza de la válvula utilizada para el control de la temperatura, la cual posee un tiempo de activación proporcional a abertura del restrictor de flujo, en cuyo intervalo el microcontrolador se dedica únicamente a determinar si ya se cumplió dicho tiempo de activación y proceder luego para tomar una muestra de la variable de entrada, esto explica el gradiente de temperatura que se observa al inicio de la acción de control, lo cual se sabe que es imposible debido a la existencia de una capacitancia térmica inherente a la variable temperatura que varía de una forma exponencial natural tanto positiva como negativamente; para el caso particular el microcontrolador 68HC12 al igual que todos los CPUs ejecutan procesos de forma secuencial y no simultáneamente, por lo tanto se trabaja con valores discretos para cada variable sobre la cual se tenga inferencia y por los motivos antes expuestos los valores discretos muestreados por el Adquisidor de Datos permanecen invariables por un periodo de tiempo considerable que no permite una apreciación a simple vista de la tendencia exponencial que exhibe la variable bajo control quedando como único recurso el hacer una abstracción mental de la tendencia antes mencionada, como puede ser para los valores donde ya se alcanzó la estabilidad, para el

caso 92° C las variaciones son mínimas por lo tanto la acción de control acondicionada a un tiempo de activación de la válvula restrictora de flujo de vapor es menor y por lo tanto las muestras de la variable bajo control son más continuas permitiendo una mejor apreciación de la tendencia con que continuaría la misma.

Para que la acción de control pudiera adecuarse a las características antes mencionadas del actuador se utilizó el circuito mostrado en la figura 3.24, puede apreciarse que la etapa digital de control se encuentra aislada eléctricamente de la etapa de potencia de DC, ello se debe a los problemas de inducción de ruido en la línea de tierra que provocaba el reset en el MCU con cada conmutación de los contactos manejadores de la señal AC para giro y alimentación del motor; mediante la señal óptica percibida por el detector se excita la base de los transistores que energizan la bobina de los relés K1, K2 y K3 utilizados para controlar la activación y la dirección de giro del motor de la válvula que controla el flujo de vapor a la centrifugadora de aceite.

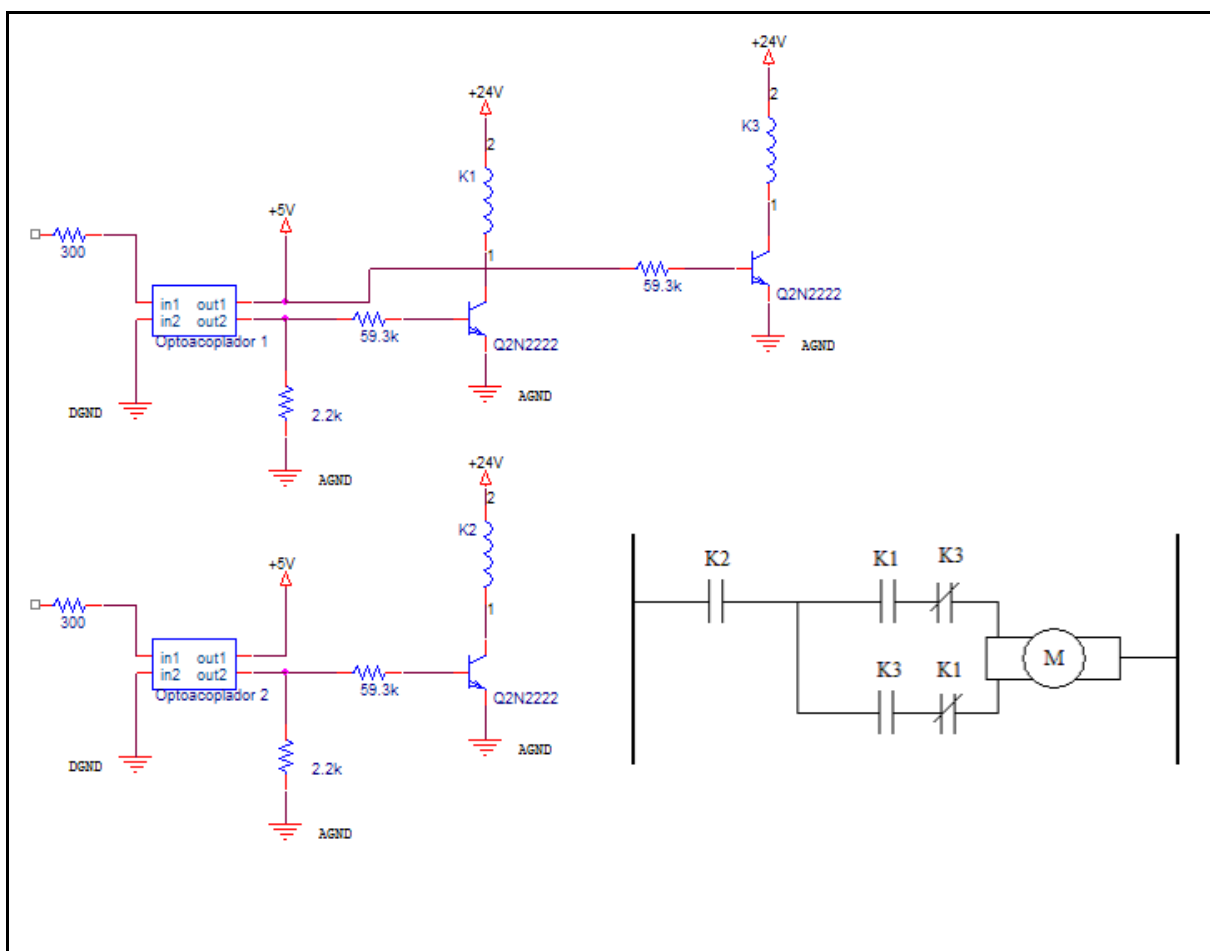


Figura 3.25. Circuito de acondicionamiento de la señal de salida del MCU68HC12 para el control de la válvula de flujo de vapor en la centrifugadora de aceite.

4.CODIGO ENSAMBLADOR PARA EL CONTROLADOR DIFUSO

A continuación se presentan los códigos en ensamblador de los programas utilizados para controlar con lógica difusa las distintas plantas en las que se desarrollaron los experimentos, cabe mencionar que los comentarios se añadieron para tener un mejor control a la hora de depurar los programas, pero que como todo controlador solo queda en la tarjeta de implementación los datos necesarios para la ejecución del programa.

4.1 CONTROL DE NIVEL DE LIQUIDO EN UN TANQUE

```
;Programa de lógica difusa
INCLUDE "macros.asm"
INCLUDE "EQUATES.ASM"

input_value:      equ $08D0
defuzz_out:       equ $08D1
setpoint:         equ $08D2
floop:           equ $08D3
términos:        equ $08D4
data:            equ $08D5
Fuzzy_inputs:    equ $08E0
Fuzzy_outputs:   equ $08EA
activate_levels: equ $08F0
MF:              equ $0900
inference_rules: equ $0930

;bloque de entradas
;setpoint
spvacio: equ Fuzzy_inputs + 0
spcasivacio: equ Fuzzy_inputs + 1
spmedio: equ Fuzzy_inputs + 2
spcasilleno: equ Fuzzy_inputs + 3
splleno: equ Fuzzy_inputs + 4
;floop
flvacio: equ Fuzzy_inputs + 5
flcasivacio: equ Fuzzy_inputs + 6
flmedio: equ Fuzzy_inputs + 7
flcasilleno: equ Fuzzy_inputs + 8
flllenno: equ Fuzzy_inputs + 9

;bloque de salidas tipo singletons
incalto: equ Fuzzy_inputs + $A
incmedio: equ Fuzzy_inputs + $B
cero: equ Fuzzy_inputs + $C
decmedio: equ Fuzzy_inputs + $D
decalto: equ Fuzzy_inputs + $E

org $08D0
```

```

dc.b $00,$00,$00,$00,$05,$0A

org $08F0 ;define los valores de activacion
dc.b $00
dc.b $40
dc.b $80
dc.b $C0
dc.b $FF

org $0900 ;define los conjuntos difusos de entrada
;setpoint
dc.b $00, $44, $00, $0F
dc.b $22, $77, $0F, $0F
dc.b $55, $AA, $0F, $0F
dc.b $88, $DD, $0F, $0F
dc.b $BB, $FF, $0F, $00

;floop
dc.b $00, $44, $00, $0F
dc.b $22, $77, $0F, $0F
dc.b $55, $AA, $0F, $0F
dc.b $88, $DD, $0F, $0F
dc.b $BB, $FF, $0F, $00

;define los offset necesarios para crear la tabla de reglas de
inferencia difusas
spvacio:      equ spvacio - Fuzzy_inputs
spcasivacio:  equ spcasivacio - Fuzzy_inputs
spmedio:      equ spmedio - Fuzzy_inputs
spcasilleno:  equ spcasilleno - Fuzzy_inputs
splleno:      equ splleno - Fuzzy_inputs
flvacio:      equ flvacio - Fuzzy_inputs
flcasivacio:  equ flcasivacio - Fuzzy_inputs
flmedio:      equ flmedio - Fuzzy_inputs
flcasilleno:  equ flcasilleno - Fuzzy_inputs
fllleno:      equ fllleno - Fuzzy_inputs
incalto:      equ incalto - Fuzzy_inputs
incmedio:     equ incmedio - Fuzzy_inputs
cero:         equ cero - Fuzzy_inputs
decmedio:     equ decmedio - Fuzzy_inputs
decalto:      equ decalto - Fuzzy_inputs

sep:          equ $FE
end_rules:    equ $FF

org $0930 ;define las reglas de inferencia difusas
dc.b spvacio, flvacio, sep, cero, sep
dc.b spvacio, flcasivacio, sep, decmedio, sep
dc.b spvacio, flmedio, sep, decalto, sep
dc.b spvacio, flcasilleno, sep, decalto, sep
dc.b spvacio, fllleno, sep, decalto, sep

dc.b spcasivacio, flvacio, sep, incmedio, sep
dc.b spcasivacio, flcasivacio, cero, sep
dc.b spcasivacio, flmedio, sep, decmedio, sep
dc.b spcasivacio, flcasilleno, sep, decalto, sep

```

```

dc.b spcasivacio, fllleno, sep, decalto, sep

dc.b spmedio, flvacio, sep, incalto, sep
dc.b spmedio, flcasivacio, incmedio, sep
dc.b spmedio, flmedio, sep, cero, sep
dc.b spmedio, flcasilleno, sep, decmedio, sep
dc.b spmedio, fllleno, sep, decalto, sep

dc.b spcasilleno, flvacio, sep, incalto, sep
dc.b spcasilleno, flcasivacio, incalto, sep
dc.b spcasilleno, flmedio, sep, incmedio, sep
dc.b spcasilleno, flcasilleno, sep, cero, sep
dc.b spcasilleno, fllleno, sep, decmedio, sep

dc.b splleno, flvacio, sep, incalto, sep
dc.b splleno, flcasivacio, incalto, sep
dc.b splleno, flmedio, sep, incalto, sep
dc.b splleno, flcasilleno, sep, incmedio, sep
dc.b splleno, fllleno, sep, cero
dc.b end_rules

;Programa Principal
org $0800

JSR INIT ; Saltar a INIT para iniciar
inicio:
JSR CONVERT ; Saltar a CONVERT para efectuar conversion
ldx #MF
ldy #Fuzzy_inputs
ldab data

Clear_inputs:
clr 1,y+
dbne b,Clear_inputs
ldy #Fuzzy_inputs

MF_eval1:
ldaa setpoint
ldab terminos
loop1:
mem
dbne b,loop1
MF_eval2:
ldaa floop
ldab terminos
cmpa #$F7
blo loop2
ldab #$ff
jmp turnoff
loop2:
mem
dbne b,loop2
ldab terminos

Clear_outputs:
clr 1,y+
dbne b,Clear_outputs

```

```

    ldx #inference_rules
    ldy #fuzzy_inputs
    ldaa #$FF
    rev

defuzz:
    ldy #fuzzy_outputs
    ldx #activate_levels
    ldab terminos
    wav
    ediv
    tfr y d
    stab defuzz_out

turnoff:
    stab PORTB
    jmp inicio

; -----
; Subrutina INIT: Iniciar ATD ;
; -----
INIT:
    LDAA #$80 ; Permite al ATD para funcion normal,
    STAA ATDCTL2 ; limpiar banderas ATD y deshabilitar interrupciones
    jsr delay ; Delay (100 uS) para tiempo de WAIT
    LDAA #$00 ; Selecccionar Modo de conversion continua en BGND
    STAA ATDCTL3 ; Ignorar FREEZE en ATDCTL3
    LDAA #$01 ; Seleccionar tiempo de muestreo final = 2 A/A de clock
    STAA ATDCTL4 ; Prescaler = Div por 4 (PRS4:0 = 1)
    movb #$FF, DDRB ; set todos los pines de PORTB como salidas
    MOVB #$34, SC0BDL; Set BAUD =9600, en SCI1 Baud Rate Reg.
    MOVB #$00, SC0CR1; Iniciar formato de datos a ocho bits,
    ; Modo de lazo y paridad deshabilitado, (SC1CR1)
    MOVB #$2C, SC0CR2; Set para Ints, y receptor habilitado(SC1CR2)
    LDAA SC0SR1; Primer paso para limpiar la bandera TDRE: Leer SC1SR1
    ldaa SC0DRL; Segundo paso para limpiar la bandera TDRE: Escribir en
    ;el registro SC1DR
    ldd #SCIsvc
    ldy #RamBase
    std UserSCI0*2,y
    cli ; clears I bit to allow interruptions
    RTS ; Return from subroutine

; -----
; Subroutine CONVERT: ;
; -----
; Set-up ATD, make single conversion and store the result to a memory
location.
; Configure and start A/D conversion
; Analog Input Signal: On PORT AD6
; Convert: using single channel, non-continuous
; The result will be located in ADR2H
CONVERT:
    LDAA #$16 ; Initializes ATD SCAN=0,MULT=1, PAD6,
    ; Write Clears Flag
    STAA ATDCTL5 ; 4 conversions on a Single Conversion
    ; sequence,

WTCONV:

```



```

    BRCLR ATDSTATH, #$80, WTCONV ; Wait for Sequence Complete Flag
    LDD ADR1H ; Loads conversion result (PAD5)
                ; into Accumulator
    STAA setpoint ; store accumulator in setpoint
    LDD ADR3H ; Loads conversion result (PAD7)
    STAA floop ; store accumulator in error
    RTS ; Return from subroutine
; * -----
; * Subroutine DELAY 100 uS *
; * -----
; Delay Required for ATD converter to Stabilize (100 uSec)
delay:
    LDAA #$C8 ; Load Accumulator with "100 uSec delay value"
wait:
    DBNE A, wait ; Decrement ACC
    RTS ; Return from subroutine
; * -----
; * Service subroutine for SCI0 interrupt
; * -----
SCIsvc:
    ldaa SC0SR1
    ldaa SC0DRL
    movb setpoint, SC0DRL
sptx:
    brclr SC0SR1, #$80, sptx
    movb floop, SC0DRL
flt看:
    brclr SC0SR1, #$80, fltx
    movb PORTB, SC0DRL
dotx:
    brclr SC0SR1, #$80, dotx
return:
    rti

```

4.2 CONTROL DE VELOCIDAD EN UN MOTOR

;Programa prueba de logica difusa

INCLUDE "macros.asm"

INCLUDE "EQUATES.ASM"

```
input_value:      equ $08D0
defuzz_out:       equ $08D1
setpoint:         equ $08D2
floop:           equ $08D3
terminos:         equ $08D4
data:             equ $08D5
Fuzzy_inputs:     equ $08E0
Fuzzy_outputs:    equ $08EA
activate_levels:  equ $08F0
MF:               equ $0900
inference_rules:  equ $0930
```

;bloque de entradas

;setpoint

```
spcero:   equ Fuzzy_inputs + 0
spmuylento:   equ Fuzzy_inputs + 1
splento:   equ Fuzzy_inputs + 2
spmedio:   equ Fuzzy_inputs + 3
sprapido:  equ Fuzzy_inputs + 4
;floop
flcero:    equ Fuzzy_inputs + 5
flmuylento:   equ Fuzzy_inputs + 6
fllento:   equ Fuzzy_inputs + 7
flmedio:   equ Fuzzy_inputs + 8
flrapido:  equ Fuzzy_inputs + 9
```

;bloque de salidas tipo singletons

```
cero:      equ Fuzzy_inputs + $A
muy poco:  equ Fuzzy_inputs + $B
poco:      equ Fuzzy_inputs + $C
medio:     equ Fuzzy_inputs + $D
todo:      equ Fuzzy_inputs + $E
```

org \$08D0

dc.b \$00,\$00,\$00,\$00,\$05,\$0A

org \$08F0 ;define los valores de activacion

dc.b \$00

dc.b \$80

dc.b \$80

dc.b \$80

dc.b \$80

org \$0900 ;define los conjuntos difusos de entrada

;setpoint

dc.b \$00, \$44, \$00, \$0F

dc.b \$22, \$77, \$0F, \$0F

dc.b \$55, \$AA, \$0F, \$0F

dc.b \$88, \$DD, \$0F, \$0F

```

dc.b $BB, $FF, $0F, $00

;floop
dc.b $00, $44, $00, $0F
dc.b $22, $77, $0F, $0F
dc.b $55, $AA, $0F, $0F
dc.b $88, $DD, $0F, $0F
dc.b $BB, $FF, $0F, $00

;define los offset necesarios para crear la tabla de reglas de
inferencia difusas
spcero:      equ spcero - Fuzzy_inputs
spmuylento:  equ spmuylento - Fuzzy_inputs
splento:     equ splento - Fuzzy_inputs
spmedio:     equ spmedio - Fuzzy_inputs
sprapido:    equ sprapido - Fuzzy_inputs
flcero:      equ flcero - Fuzzy_inputs
flmuylento:  equ flmuylento - Fuzzy_inputs
fllento:     equ fllento - Fuzzy_inputs
flmedio:     equ flmedio - Fuzzy_inputs
flrapido:    equ flrapido - Fuzzy_inputs
cero:        equ cero - Fuzzy_inputs
muypoco:     equ muypoco - Fuzzy_inputs
poco:        equ poco - Fuzzy_inputs
medio:       equ medio - Fuzzy_inputs
todo:        equ todo - Fuzzy_inputs

sep:         equ $FE
end_rules:    equ $FF

org $0930 ;define las reglas de inferencia difusas
dc.b spcero, flcero, sep, cero, sep
dc.b spcero, flmuylento, sep, cero, sep
dc.b spcero, fllento, sep, cero, sep
dc.b spcero, flmedio, sep, cero, sep
dc.b spcero, flrapido, sep, cero, sep

dc.b spmuylento, flcero, sep, muypoco, sep
dc.b spmuylento, flmuylento, sep, muypoco, sep
dc.b spmuylento, fllento, sep, cero, sep
dc.b spmuylento, flmedio, sep, cero, sep
dc.b spmuylento, flrapido, sep, cero, sep

dc.b splento, flcero, sep, poco, sep
dc.b splento, flmuylento, sep, poco, sep
dc.b splento, fllento, sep, poco, sep
dc.b splento, flmedio, sep, cero, sep
dc.b splento, flrapido, sep, cero, sep

dc.b spmedio, flcero, sep, medio, sep
dc.b spmedio, flmuylento, sep, medio, sep
dc.b spmedio, fllento, sep, medio, sep
dc.b spmedio, flmedio, sep, medio, sep
dc.b spmedio, flrapido, sep, cero, sep

dc.b sprapido, flcero, sep, todo, sep
dc.b sprapido, flmuylento, sep, todo, sep

```

```

    dc.b sprapido, flento, sep, todo, sep
    dc.b sprapido, flmedio, sep, todo, sep
    dc.b sprapido, flrapido, sep, todo
    dc.b end_rules

;Main program
    org $0800

    JSR INIT ; Branch to INIT subroutine to Initialize ATD
inicio:
    JSR CONVERT ; Branch to CONVERT Subroutine for conversion
    ldx #MF
    ldy #Fuzzy_inputs
    ldab data

Clear_inputs:
    clr 1,y+
    dbne b,Clear_inputs
    ldy #Fuzzy_inputs

MF_eval1:
    ldaa setpoint
    ldab terminos
loop1:
    mem
    dbne b,loop1
MF_eval2:
    ldaa floop
    ldab terminos
    cmpa #$F7
    blo loop2
    ldab #$00
    jmp turnoff
loop2:
    mem
    dbne b,loop2
    ldab terminos

Clear_outputs:
    clr 1,y+
    dbne b,Clear_outputs
    ldx #inference_rules
    ldy #fuzzy_inputs
    ldaa #$FF
    rev

defuzz:
    ldy #fuzzy_outputs
    ldx #activate_levels
    ldab terminos
    wav
    ediv
    tfr y d
    stab defuzz_out

turnoff:
    stab PORTB

```

```

        jmp inicio

; -----
; Subroutine INIT: Initialize ATD ;
; -----
INIT:
    LDAA #$80 ; Allow ATD to function normally,
    STAA ATDCTL2 ; ATD Flags clear normally & disable interrupts
    jsr delay ; Delay (100 uS) for WAIT delay time.
    LDAA #$00 ; Select continue conversion in BGND Mode
    STAA ATDCTL3 ; Ignore FREEZE in ATDCTL3
    LDAA #$01 ; Select Final Sample time = 2 A/D clocks
    STAA ATDCTL4 ; Prescaler = Div by 4 (PRS4:0 = 1)
    movb #$FF, DDRB ; sets all PORTB pins as outputs
    MOVB #$34, SC0BDL; Set BAUD =9600, in SCI1 Baud Rate Reg.
    MOVB #$00, SC0CR1; Initialize for 8-bit Data format,
                    ; Loop Mode and parity disabled, (SC1CR1)
    MOVB #$2C, SC0CR2; Set for Ints, and reciever enabled(SC1CR2)
    LDAA SC0SR1 ; 1st step to clear TDRE flag: Read SC1SR1
    ldaa SC0DRL ; 2nd step to clear TDRE flag: Write SC1DR register
    ldd #SCIsvc
    ldy #RamBase
    std UserSCI0*2,y
    cli ; clears I bit to allow interruptions
    RTS ; Return from subroutine

; -----
; Subroutine CONVERT: ;
; -----
; Set-up ATD, make single conversion and store the result to a memory
location.
; Configure and start A/D conversion
; Analog Input Signal: On PORT AD6
; Convert: using single channel, non-continuous
; The result will be located in ADR2H
CONVERT:
    LDAA #$16 ; Initializes ATD SCAN=0,MULT=1, PAD6,
                ; Write Clears Flag
    STAA ATDCTL5 ; 4 conversions on a Single Conversion
                ; sequence,

WTCONV:
    BRCLR ATDSTATH,$$80,WTCONV ; Wait for Sequence Complete Flag
    LDD ADR1H ; Loads conversion result(PAD5)
                ; into Accumulator
    STAA setpoint ;store accumulator in setpoint
    LDD ADR3H ;Loads conversion result (PAD7)
    STAA flop ; store accumulator in error
    RTS ; Return from subroutine

;* -----
;* Subroutine DELAY 100 uS *
;* -----
; Delay Required for ATD converter to Stabilize (100 uSec)
delay:
    LDAA #$C8 ; Load Accumulator with "100 uSec delay value"
wait:
    DBNE A, wait ; Decrement ACC
    RTS ; Return from subroutine
;* -----

```

```

;* Service subroutine for SCIO interrupt
;* -----
SCIsvc:
    ldaa SC0SR1
    ldaa SC0DRL
    movb setpoint, SC0DRL
sptx:
    brclr SC0SR1, # $80, sptx
    movb floop, SC0DRL
flt看:
    brclr SC0SR1, # $80, fltx
    movb PORTB, SC0DRL
dotx:
    brclr SC0SR1, # $80, dotx
return:
    rti

```

4.3 CONTROL DE TEMPERATURA EN UNA CENTRIFUGADORA DE ACEITE

```

;Programa de logica difusa
INCLUDE "macros.asm"
INCLUDE "EQUATES.ASM"

org $0930
temperatura:      dc.b $00
defuzz_out:       dc.b $00
counter:          dc.b $00
salida:           dc.b $00
terminos:         dc.b $03
vueltas:          dc.b $00
anterior:         dc.b $00
actual:           dc.b $00

Fuzzy_inputs:     equ $0940
Fuzzy_outputs:    equ $0943
activate_levels:  equ $0950
MF:               equ $0960
inference_rules:  equ $0970

;bloque de entradas
;temperatura
baja:             equ Fuzzy_inputs + $0
optima:           equ Fuzzy_inputs + $1
alta:             equ Fuzzy_inputs + $2

;bloque de salidas tipo singletons
parar:            equ Fuzzy_inputs + $3
abrir_medio:      equ Fuzzy_inputs + $4
abrir_todo:       equ Fuzzy_inputs + $5

org $0950 ;define los valores de activacion
dc.b $00

```

```

dc.b $FF
dc.b $FF

org $0960 ;define los conjuntos difusos de entrada
;error
dc.b $00, $A0, $00, $0F
dc.b      $96, $AB, $33, $33
dc.b      $A0, $FF, $0F, $00

;define los offset necesarios para crear la tabla de reglas de
inferencia difusas
temp_baja:      equ baja - Fuzzy_inputs
temp_optima:    equ optima - Fuzzy_inputs
temp_alta:      equ alta - Fuzzy_inputs
parar:          equ parar - Fuzzy_inputs
midopen:        equ abrir_medio - Fuzzy_inputs
open:           equ abrir_todo - Fuzzy_inputs

sep:            equ $FE
end_rules:      equ $FF

org $0970 ;define las reglas de inferencia difusas
dc.b temp_baja, sep, open, sep
dc.b temp_optima, sep, parar, sep
dc.b temp_alta, sep, midopen
dc.b end_rules

;Main program
org $0800

JSR INIT  ; Branch to INIT subroutine to Initialize ATD
inicio:
JSR CONVERT  ; Branch to CONVERT Subroutine for conversion
ldx #MF
ldy #Fuzzy_inputs

MF_eval1:
ldaa temperatura
ldab terminos
loop1:
mem
dbne b,loop1
ldab terminos

Clear_outputs:
clr 1,y+
dbne b,Clear_outputs
ldx #inference_rules
ldy #fuzzy_inputs
ldaa #$FF
rev

defuzz:
ldy #fuzzy_outputs
ldx #activate_levels
ldab terminos
wav

```

```

    ediv
    tfr y d
    stab defuzz_out
    beq return
    cmpb anterior
    beq return
    blo giro_cerrar
    ldaa defuzz_out
    ldab anterior
    sba
    tab
    ldaa #$40

outport:
    staa PORTB
    stab vueltas
hold:
    LDAA TFLG1 ; To clear OC2 Flag, first it must be read,
    movb #$04,TFLG1
WTF LG:
    BRCLR TFLG1,$$02,WTF LG ; Wait (Polling) for C2F Flag
    LDD TC1H ; Loads value of compare from TC2 Reg.
    ADDD #$E160 ; Add hex value of 500us High Time
    STD TC1H ; Set-up next transition time in 500 us
    ldab vueltas
    cmpb counter
    bhi hold

return:
    movb defuzz_out,anterior
    movb #$00,PORTB
    clr counter
    clr vueltas
    jmp inicio

giro_cerrar:
    ldaa anterior
    sba
    tab
    ldaa #$C0
    bra outport

; -----
; Subroutine INIT: Initialize ATD ;
; -----
INIT:
    LDAA #$80 ; Allow ATD to function normally,
    STAA ATDCTL2 ; ATD Flags clear normally & disable interrupts
    jsr delay ; Delay (100 uS) for WAIT delay time.
    LDAA #$00 ; Select continue conversion in BGND Mode
    STAA ATDCTL3 ; Ignore FREEZE in ATDCTL3
    LDAA #$01 ; Select Final Sample time = 2 A/D clocks
    STAA ATDCTL4 ; Prescaler = Div by 4 (PRS4:0 = 1)
    movb #$FF, DDRB ; sets all PORTB pins as outputs
    ldd #SCIsvc
    ldy #RamBase
    std UserSCI0*2,y
    MOVb #$34,SC0BDL; Set BAUD =9600, in SCI1 Baud Rate Reg.

```



```

MOVW #$00,SC0CR1; Initialize for 8-bit Data format,
        ; Loop Mode and parity disabled,(SC1CR1)
MOVW #$2C,SC0CR2; Set for Ints, and reciever enabled(SC1CR2)
LDAA SC0SR1      ; 1st step to clear TDRE flag: Read SC1SR1
ldaa SC0DRL      ; 2nd step to clear TDRE flag: Write SC1DR register
ldd #PACCsvc
ldy #RamBase
std UserPAccEdge*2,y
clr TMSK1
MOVW #$25,TMSK2      ; Disable overflow interrupt, disable pull-up
        ; resistor function with normal drive capability
        ; and free running counter, Prescaler = sys clock/32.
MOVW #$90,TSCR ; Enable Timer, Timer runs during wait state, and
        ; while in Background Mode, also clear flags
        ; normally.
MOVW #$04,TCTL2 ; Initialize OC2 to toggle on successful compare.
MOVW #$02,TIOS ; Select Channel 2 to act as output compare.
MOVW #$E160,TC1H ; Load TC2 Reg with initial compare value.
MOVW #$51,PACTL      ; Habilita el PAC, y la interrupcion
ldd PACNT
cli          ; clears I bit to allow interruptions
RTS          ; Return from subroutine
; -----
; Subroutine CONVERT: ;
; -----
; Set-up ATD, make single conversion and store the result to a memory
location.
; Configure and start A/D conversion
; Analog Input Signal: On PORT AD6
; Convert: using single channel, non-continuous
; The result will be located in ADR2H
CONVERT:
    LDAA #$16 ; Initializes ATD SCAN=0,MULT=1, PAD6,
        ; Write Clears Flag
    STAA ATDCTL5 ; 4 conversions on a Single Conversion
        ; sequence,
WTCONV:
    BRCLR ATDSTATH,$$80,WTCONV ; Wait for Sequence Complete Flag
    LDD ADR3H ; Loads conversion result(PAD7)
        ; into Accumulator
    STAA temperatura ;store accumulator in setpoint
;    LDD ADR3H ;Loads conversion result (PAD7)
;    STAA salida ; store accumulator in error
    RTS          ; Return from subroutine
;* -----
;* Subroutine DELAY 100 uS *
;* -----
; Delay Required for ATD converter to Stabilize (100 uSec)
delay:
    LDAA #$C8 ; Load Accumulator with "100 uSec delay value"
wait:
    DBNE A, wait ; Decrement ACC
    RTS          ; Return from subroutine
;* -----
;* Service subroutine for SCI0 interrupt
;* -----
SCIsvc:

```

```

        ldaa SC0SR1
        ldaa SC0DRL
        movb temperatura,SC0DRL
sptx:
        brclr SC0SR1,#$80,sptx
        rti
;* Service subroutine for PACC interrupt
;* -----
PACCsvc:
        ;* To clear the C2F flag: 1) read TFLG1 when
        ;* C2F is set and then 2) write a logic "one" to C2F.
        LDD PACNT
        stab counter
        rti

```

5. ALCANCES Y LIMITACIONES.

El Simulador Difuso tiene la capacidad de manejar hasta cinco entradas cada una de las cuales puede estar compuesta por cinco términos o conjuntos difusos.

Las entradas deben ser acondicionadas por medio de circuitos externos para poder obtener una señal de voltaje de 0 a 5Vdc, lo cual está supedita a que las entradas del microcontrolador seleccionado para el desarrollo del presente trabajo de graduación solo posee entradas de voltaje de 0 a 5 Vdc para el convertidor de análogo a digital.

Las representaciones gráficas de los sistemas difusos no pueden ser abiertas ni editadas una vez se cierra el programa, lo que si queda guardado es el archivo ASM generado para el controlador en el MCU 68HC12 así como los datos que resultan de las simulaciones con las variables, salidas y reglas definidas.

6. RECOMENDACIONES

Ante una eventual comercialización de la aplicación del Simulador Difuso se sugiere el diseño y construcción de un acondicionador de señal genérico para transductores estandarizados, es decir, que sea fácil de interconectar con los transductores encontrados en el mercado de dispositivos para Instrumentación y Control; con una entrada para Pt-100, una para termocupla, una entrada de corriente de 4 a 20 mA y una de voltaje de 0 a 10 V se obtendría una versatilidad amplia del sistema.

Se recomienda además el incluir en el programa la opción de guardar las representaciones gráficas de los sistemas simulados, para que una vez cerrados puedan ser abiertos de nuevo por el usuario para editarlos o para realizar de nuevo las simulaciones.

7.CONCLUSIONES

1. Por medio de la implementación de un Controlador Basado en lógica difusa puede obtenerse una respuesta en la planta a controlar que puede representar una reducción en los costos de adquisición como de mantenimiento tanto preventivo como correctivo, si se toma como referencia los controladores comercializados en la actualidad.
2. En un Controlador Difuso es de vital importancia contar con la colaboración de una persona, el ente experto, para la definición de las reglas difusas que el sistema va a ejecutar para mantener bajo control las variables que requieren control automático.
3. Dependiendo de la variable o variables a controlar por el sistema Difuso, se hace necesario el diseño y la implementación de acondicionadores de señal tanto para la entrada, proveniente del sensor, como de la etapa de potencia, la cual se conecta al actuador. Cabe mencionar que para sensores estandarizados industriales se puede utilizar entradas genéricas como lo son entradas de corriente de 4 a 20 mA o entradas de voltaje de 0 a 10V, ya que éstas son salidas provistas por muchos transductores presentes en el mercado de dispositivos de instrumentación y control.
4. Con la Aplicación del Simulador Difuso se proporciona una herramienta a la persona encargada de la implementación de un Sistema de control ya que le brinda una interfaz grafica amigable para tener una mejor perspectiva tanto de la

- definición de los conjuntos difusos como de la congruencia de las reglas de inferencia difusa.
5. Cuando se implementa un Sistema de Control en un ambiente Industrial, deben tomarse en consideración aspectos del ambiente ajenos al controlador como lo son fuentes de ruido magnético, vibración, temperaturas extremas, humedad. Deben tomarse las medidas de seguridad pertinentes para eliminar o reducir a límites aceptables dichas condiciones por medio de apantallamiento de cables, uso adecuado de la tierra de protección como de la conexión a tierra de los equipos de baja intensidad.
 6. Aprovechando la versatilidad que provee el Microcontrolador Motorola MCU68HC12 se implementó un Sistema de Control Difuso basado en el soporte que dicho microcontrolador proporciona para el desarrollo de aplicaciones de lógica difusa.
 7. El uso de interrupciones es una herramienta de mucha utilidad, de la cual carecen muchos controladores comerciales, y puede utilizarse para realizar tareas de servicio ante eventos que requieran acción inmediata ya sea por parte del sistema o por parte de personal técnico alertado por el sistema.
 8. Al desarrollar la aplicación en una plataforma ampliamente difundida como lo es Visual Basic, se obtuvo un aporte de herramientas amplio incrementado también por el uso del manejo de errores que dicho lenguaje de programación proporciona al usuario.

8. BIBLIOGRAFÍA

- [1] NILSSON Nils. J Inteligencia Artificial Una Nueva Síntesis, Mc Graw Hill Impreso en España en 2001. Primera Edición
- [2] MENDOZA, José. García, Oscar. Aguilar, Walter. Diseño de prototipo de un controlador Inteligente Basado en Lógica Difusa para controlar las plantas de laboratorio de Automatización del CITT de la Universidad Don Bosco. 1999.
- [3] FLEB, a Fuzzy logic E-Book, Pedro Argueta, www.esime.edu.es.
- [4] CPU12 Reference Manual, Motorola semiconductors, 2002.
- [5] M68EVB912BC32 Evaluation Board User's Manual, Motorola Inc. Semiconductors Division Octubre 1997.
- [6] MC68HC12 Advanced Information, Motorola Semiconductors, 2002.
- [7] PEDRIC, Witold. FUZZY CONTROL AND FUZZY SYSTEMS. Research Studies. England. 1999
- [9] *Cortés Mateos, Raúl* "Tesis Doctoral" SEPI-ESIME, 1997.
- [10] *Cady, F. M., y J. M. Sibigtroth* Software and Hardware Engineering: MOTOROLA M68HC12, Oxford U. Press, 2000.
- [11] **MOTOROLA** M68HC12 Reference Manual, 1998.
- [12] *Mamdani, E. H. "y S. Assilian* "An experiment in linguistic síntesis with a fuzzy logic controller", Int. J. Man-Mach. Stud., 7(1) pp 1-13, 1975.
- [13] *Lin, Chin-Teng y C.S.G. Lee* NEURAL FUZZY SYSTEMS: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice Hall P T R, 1996.
- [14] *Dubois, D., y H. Prade* "Unfair Coins and necessity measures: Toward a possibilistic interpretations of histograms", Fuzzy Sets Sist. 10(1) pp 15-20, 1985.

- [15] **Larsen, P. M.**, “Industrial Application of fuzzy logic control”, Int. J. Man-Mach. Stud.,
12(1) pp 3-10, 1980.
- [16] **Zadeh, L. A.** “Outline of a new approach to the analysis of complex systems and decision
proceses”, IEEE Trans. on Sist. Man and Cyber. SMC-1 pp 28-44, 1973.
- [17] **Kaufmann, A.**, Introduction to the Theory of Fuzzy Subsets, NY: Academic Press,
1975.
- [18] **Mizumoto, M** “Fuzzy Sets and their Operations”, Inf. Control 48 pp 30-48, 1981.
- [19] **Mizumoto, M.** “Fuzzy controls under various fuzzy reasoning methods”, Inf. Sci. 45
pp
129-152, 1988.
- [20] **Sugeno, M., y K. Tanaka** “Successive identification of fuzzy model and its
application to prediction of complex systems”, Fuzzy Sets Sist., 42 pp 315-344, 1991.

APENDICES

**A.1 CODIGO FUENTE DE LA
APLICACIÓN DEL SIMULADOR
DIFUSO EN VISUAL BASIC**

Dim lazo, i As Integer

Dim term As String

Option Explicit

Private Sub CancelButton_Click()

ee.Hide

End Sub

Private Sub OKButton_Click()

With frmTesis

input_index = CInt(Val(txtinput))

input_index = input_index + 0

rb_index = CInt(Val(txtruleblock))

rb_index = rb_index + 6

output_index = CInt(Val(txtoutput))

output_index = output_index + 9

If rb_index < 7 Then

MsgBox "Debe agregar al menos un bloque de reglas", vbExclamation, "Atención"

GoTo ignoreaceptar

End If

If input_index < 1 Then

MsgBox "El sistema necesita al menos una entrada", vbExclamation, "Atención"

GoTo ignoreaceptar

End If

If output_index < 10 Then

MsgBox "El sistema necesita al menos una salida", vbExclamation, "Atención"

GoTo ignoreaceptar

End If

If input_index > 5 Then

MsgBox "Se excedió número máximo de entradas", vbExclamation, "Atención"

GoTo ignoreaceptar

End If

If rb_index > 8 Then

MsgBox "Se excedió número máximo de bloques de reglas", vbExclamation, "Atención"

GoTo ignoreaceptar

End If

If output_index > 12 Then

```

MsgBox "Se excedió número máximo de salidas", vbExclamation, "Atención"
GoTo ignoreacceptar
End If

For lazo = 7 To rb_index
    .Picture1(lazo).Left = 3392
    .Picture1(lazo).Top = (lazo - 7) * .Picture1(7).Height + (lazo - 6) * (.Frame1.Height - (rb_index - 6) *
    (.Picture1(7).Height)) / (rb_index - 5)
    .Picture1(lazo).Picture = LoadPicture("C:\Documents and Settings\Khalena\Mis
documentos\Tesis\Programas VB\ruleblock.bmp")
Next lazo

If rb_index = 8 Then
    For lazo = 1 To input_index
        If .Line1(lazo).Visible = True Then
            .Line1(lazo).X1 = .Picture1(lazo).Left + .Picture1(lazo).Width
            .Line1(lazo).X2 = .Picture1(7).Left
            .Line1(lazo).Y1 = .Picture1(lazo).Top + Round((.Picture1(lazo).Height / 2), 0)
            .Line1(lazo).Y2 = .Picture1(7).Top + Round((.Picture1(7).Height / 2), 0)
        End If
    Next lazo

    For lazo = 10 To output_index
        If .Line1(lazo).Visible = True Then
            .Line1(lazo).X1 = .Picture1(lazo).Left
            .Line1(lazo).X2 = .Picture1(7).Left + .Picture1(7).Width
            .Line1(lazo).Y1 = .Picture1(lazo).Top + Round((.Picture1(lazo).Height / 2), 0)
            .Line1(lazo).Y2 = .Picture1(7).Top + Round((.Picture1(7).Height / 2), 0)
        End If
    Next lazo
End If

For lazo = 1 To input_index
    .Picture1(lazo).Left = 886
    .Picture1(lazo).Top = (lazo - 1) * .Picture1(1).Height + lazo * (.Frame1.Height - input_index *
    (.Picture1(1).Height)) / (input_index + 1)
    .Picture1(lazo).Picture = LoadPicture("C:\Documents and Settings\Khalena\Mis
documentos\Tesis\Programas VB\input.bmp")
Next lazo

```

```

If rb_index = 7 Then
    For lazo = 1 To input_index
        .Line1(lazo).X1 = .Picture1(lazo).Left + .Picture1(lazo).Width
        .Line1(lazo).X2 = .Picture1(7).Left
        .Line1(lazo).Y1 = .Picture1(lazo).Top + Round((.Picture1(lazo).Height / 2), 0)
        .Line1(lazo).Y2 = .Picture1(7).Top + Round((.Picture1(7).Height / 2), 0)
        .Line1(lazo).Visible = True
    Next lazo
Else
    For lazo = 1 To input_index
        If .Line1(lazo).Visible = True Then
            .Line1(lazo).X1 = .Picture1(lazo).Left + .Picture1(lazo).Width
            .Line1(lazo).X2 = .Picture1(7).Left
            .Line1(lazo).Y1 = .Picture1(lazo).Top + Round((.Picture1(lazo).Height / 2), 0)
            .Line1(lazo).Y2 = .Picture1(7).Top + Round((.Picture1(7).Height / 2), 0)
        End If
    Next
End If

For lazo = 10 To output_index
    .Picture1(lazo).Left = 6565
    .Picture1(lazo).Top = (lazo - 10) * .Picture1(10).Height + (lazo - 9) * (.Frame1.Height - (output_index - 9) * (.Picture1(10).Height)) / (output_index - 8)
    .Picture1(lazo).Picture = LoadPicture("C:\Documents and Settings\Khalena\Mis documentos\Tesis\Programas VB\output.bmp")
Next lazo

If rb_index = 7 Then
    For lazo = 10 To output_index
        .Line1(lazo).X1 = .Picture1(lazo).Left
        .Line1(lazo).X2 = .Picture1(7).Left + .Picture1(7).Width
        .Line1(lazo).Y1 = .Picture1(lazo).Top + Round((.Picture1(lazo).Height / 2), 0)
        .Line1(lazo).Y2 = .Picture1(7).Top + Round((.Picture1(7).Height / 2), 0)
        .Line1(lazo).Visible = True
    Next lazo
Else
    For lazo = 10 To output_index
        .Line1(lazo).X1 = .Picture1(lazo).Left

```

```

        .Line1(lazo).X2 = .Picture1(7).Left + .Picture1(7).Width
        .Line1(lazo).Y1 = .Picture1(lazo).Top + Round((.Picture1(lazo).Height / 2), 0)
        .Line1(lazo).Y2 = .Picture1(7).Top + Round((.Picture1(7).Height / 2), 0)
    Next lazo
End If

Text1.Text = txtinput + txtoutput + txtruleblock
End With
For lazo = 1 To input_index
    matriz_indices(lazo, 2) = 3
    matriz_indices(lazo, 1) = lazo
Next lazo
For i = 1 To input_index
    If i = 1 Then
        For lazo = 1 To matriz_indices(1, 2)
            frminput1.Lstterminos.AddItem "term" + Str(lazo)
            frmruleblock1.Combo1(i).AddItem "term" + Str(lazo)
        Next lazo
        VarNames(i) = "Variable " + Str(i)
    End If
    If i = 2 Then
        For lazo = 1 To matriz_indices(2, 2)
            frminput2.Lstterminos.AddItem "term" + Str(lazo)
            frmruleblock1.Combo1(i).AddItem "term" + Str(lazo)
        Next lazo
        VarNames(i) = "Variable " + Str(i)
    End If
    If i = 3 Then
        For lazo = 1 To matriz_indices(3, 2)
            frminput3.Lstterminos.AddItem "term" + Str(lazo)
            frmruleblock1.Combo1(i).AddItem "term" + Str(lazo)
        Next lazo
        VarNames(i) = "Variable " + Str(i)
    End If
    If i = 4 Then
        For lazo = 1 To matriz_indices(4, 2)
            frminput4.Lstterminos.AddItem "term" + Str(lazo)

```

```

        frmruleblock1.Combo1(i).AddItem "term" + Str(lazo)
    Next lazo
    VarNames(i) = "Variable " + Str(i)
End If
If i = 5 Then
    For lazo = 1 To matriz_indices(5, 2)
        frminput5.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo1(i).AddItem "term" + Str(lazo)
    Next lazo
    VarNames(i) = "Variable " + Str(i)
End If
Next i
For lazo = 1 To output_index - 9
    matriz_indices(lazo, 4) = 3
    matriz_indices(lazo, 3) = lazo
Next lazo

For i = 1 To output_index - 9
    If i = 1 Then
        For lazo = 1 To matriz_indices(1, 4)
            frmoutput1.Lstterminos.AddItem "term" + Str(lazo)
            frmruleblock1.Combo6(i).AddItem "term" + Str(lazo)
        Next lazo
        OutNames(i) = "Salida " + Str(i)
    End If
    If i = 2 Then
        For lazo = 1 To matriz_indices(2, 4)
            frmoutput2.Lstterminos.AddItem "term" + Str(lazo)
            frmruleblock1.Combo6(i).AddItem "term" + Str(lazo)
        Next lazo
        OutNames(i) = "Salida " + Str(i)
    End If
    If i = 3 Then
        For lazo = 1 To matriz_indices(3, 4)
            frmoutput3.Lstterminos.AddItem "term" + Str(lazo)
            frmruleblock1.Combo6(i).AddItem "term" + Str(lazo)
        Next lazo
    End If

```



```

        OutNames(i) = "Salida " + Str(i)
    End If
Next i

```

```

ee.Visible = False
ignoreacceptar:
End Sub

```

```

Dim xpos, ypos As Double
Dim xmin, xmax, gap As Double
Dim y, X1, X2, X3, X4 As Byte

```

```

Private Sub Form_Activate()
Call Text2_Change
Call Text3_Change
End Sub

```

```

Private Sub Form_Click()
For lazo = 1 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End Sub

```

```

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
ypos = (3120 - y) / 3000
If x > 120 And x < 5120 And y > 120 And y < 3120 Then
    Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
End If
End Sub

```

```

Private Sub cmdacceptarinput1_Click()
For lazo = 0 To Lstterminos.ListCount - 1
    frmruleblock1.Combo1(1).List(lazo) = Lstterminos.List(lazo)

```

```

Next lazo
frminput1.Hide
End Sub

```

```

Private Sub Lstterminos_Click()

```

```

If Lstterminos.ListIndex = 0 Then
    For lazo = 1 To 4
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 5 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If

```

```

If Lstterminos.ListIndex = 1 Then
    For lazo = 1 To 4
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 5 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 9 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If

```

```

If Lstterminos.ListIndex = 2 Then
    For lazo = 1 To 8
        If Line17(lazo).Visible = True Then

```

```

        Line17(lazo).BorderWidth = 1
    End If
Next lazo
For lazo = 9 To 12
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 2
    End If
Next lazo
For lazo = 13 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 3 Then
    For lazo = 1 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 13 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo

```

```

For lazo = 17 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 2
    End If
Next lazo
End If
End Sub

```

```

Private Sub Lstterminos_DblClick()

```

```

    If Lstterminos.ListIndex = 0 Then
        frmtermino1prop.Show
    End If

```

```

    If Lstterminos.ListIndex = 1 Then
        frmtermino2prop.Show
    End If

```

```

    If Lstterminos.ListIndex = 2 Then
        frmtermino3prop.Show
    End If

```

```

    If Lstterminos.ListIndex = 3 Then
        frmtermino4prop.Show
    End If

```

```

    If Lstterminos.ListIndex = 4 Then
        frmtermino5prop.Show
    End If

```

```

End Sub

```

```

Private Sub tbrdefin1_ButtonClick(ByVal Button As MSComctlLib.Button)

```

```

    Select Case Button.Key
        Case "addterm"
            matriz_indices(1, 2) = matriz_indices(1, 2) + 1
            If Lstterminos.ListCount = 3 Then
                Lstterminos.AddItem "term 4"
            End If
        Case Else
            'Do Nothing
        End Select

```

```

frmruleblock1.Combo1(1).AddItem "term" + Str(4)
frmtermino4prop.Show
Else
    If Lstterminos.ListCount = 4 Then
        Lstterminos.AddItem "term 5"
        frmruleblock1.Combo1(1).AddItem "term" + Str(5)
        frmtermino5prop.Show
    Else
        matriz_indices(1, 2) = 5
        MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"
    End If
End If
End Select
End Sub

```

```

Private Sub Text1_Change()
    VarNames(1) = Text1.Text
    frminput1.Caption = "Definición de Entrada " + VarNames(1)
End Sub

```

```

Private Sub Text2_Change()
    xmin = Val(Text2.Text)
    Universo_entrada(1, 1) = xmin
    gap = (xmax - xmin) / 5
    For lazo = 0 To 5
        Text4(lazo + 1) = Str(xmin + gap * lazo)
    Next lazo
    frmtermino1prop.Text5.Text = Universo_entrada(1, 1)
End Sub

```

```

Private Sub Text3_Change()
    xmax = Val(Text3.Text)
    Universo_entrada(1, 2) = xmax
    gap = (xmax - xmin) / 5
    For lazo = 0 To 5
        Text4(lazo + 1) = Str(xmin + gap * lazo)
    Next lazo

```

```
frmtermino3prop.Text8.Text = Universo_entrada(1, 2)
```

```
End Sub
```

```
Dim xpos, ypos As Double
```

```
Dim xmin, xmax, gap As Double
```

```
Private Sub Form_Activate()
```

```
Call Text2_Change
```

```
Call Text3_Change
```

```
End Sub
```

```
Private Sub Form_Click()
```

```
For lazo = 1 To 20
```

```
    If Line17(lazo).Visible = True Then
```

```
        Line17(lazo).BorderWidth = 1
```

```
    End If
```

```
Next lazo
```

```
End Sub
```

```
Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
```

```
xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
```

```
ypos = (3120 - y) / 3000
```

```
If x > 120 And x < 5120 And y > 120 And y < 3120 Then
```

```
    Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
```

```
End If
```

```
End Sub
```

```
Private Sub cmdaceptarinput2_Click()
```

```
For lazo = 0 To Lstterminos.ListCount - 1
```

```
    frmruleblock1.Combo1(2).List(lazo) = Lstterminos.List(lazo)
```

```
Next lazo
```

```
frminput2.Hide
```

```
End Sub
```

```
Private Sub Lstterminos_Click()
```

```
If Lstterminos.ListIndex = 0 Then
```

```
    For lazo = 1 To 4
```

```
If Line17(lazo).Visible = True Then
    Line17(lazo).BorderWidth = 2
End If
Next lazo
For lazo = 5 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 1 Then
    For lazo = 1 To 4
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 5 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 9 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 2 Then
    For lazo = 1 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 9 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
End If
```

```

Next lazo
For lazo = 13 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 3 Then
    For lazo = 1 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 13 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
End If
End Sub

```



```
Private Sub Lstterminos_DblClick()
```

```
    If Lstterminos.ListIndex = 0 Then
```

```
        frmtermino1prop2.Show
```

```
    End If
```

```
    If Lstterminos.ListIndex = 1 Then
```

```
        frmtermino2prop2.Show
```

```
    End If
```

```
    If Lstterminos.ListIndex = 2 Then
```

```
        frmtermino3prop2.Show
```

```
    End If
```

```
    If Lstterminos.ListIndex = 3 Then
```

```
        frmtermino4prop2.Show
```

```
    End If
```

```
    If Lstterminos.ListIndex = 4 Then
```

```
        frmtermino5prop2.Show
```

```
    End If
```

```
End Sub
```

```
Private Sub tbrdefin2_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
    Select Case Button.Key
```

```
        Case "addterm"
```

```
            matriz_indices(2, 2) = matriz_indices(2, 2) + 1
```

```
            If Lstterminos.ListCount = 3 Then
```

```
                Lstterminos.AddItem "term" + Str(4)
```

```
                frmruleblock1.Combo1(2).AddItem "term" + Str(4)
```

```
                frmtermino4prop2.Show
```

```
            Else
```

```
                If Lstterminos.ListCount = 4 Then
```

```
                    Lstterminos.AddItem "term" + Str(5)
```

```
                    frmruleblock1.Combo1(2).AddItem "term" + Str(5)
```

```
                    frmtermino5prop2.Show
```

```

Else
    matriz_indices(2, 2) = 5
    MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"
End If
End If
End Select
End Sub

```

```

Private Sub Text1_Change()
    VarNames(2) = Text1.Text
    frminput2.Caption = "Definición de Entrada " + VarNames(2)
End Sub

```

```

Private Sub Text2_Change()
    xmin = Val(Text2.Text)
    Universo_entrada(2, 1) = xmin
    gap = (xmax - xmin) / 5
    For lazo = 0 To 5
        Text4(lazo + 1) = Str(xmin + gap * lazo)
    Next lazo
    frmtermino1prop2.Text5.Text = Universo_entrada(2, 1)
End Sub

```

```

Private Sub Text3_Change()
    xmax = Val(Text3.Text)
    Universo_entrada(2, 2) = xmax
    gap = (xmax - xmin) / 5
    For lazo = 0 To 5
        Text4(lazo + 1) = Str(xmin + gap * lazo)
    Next lazo
    frmtermino3prop2.Text8.Text = Universo_entrada(2, 2)
End Sub

```

```

Dim xpos, ypos As Double
Dim xmin, xmax, gap As Double

```

```
Private Sub Form_Activate()
```

```
    Call Text2_Change
```

```
    Call Text3_Change
```

```
End Sub
```

```
Private Sub Form_Click()
```

```
    For lazo = 1 To 20
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 1
```

```
        End If
```

```
    Next lazo
```

```
End Sub
```

```
Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
```

```
    xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
```

```
    ypos = (3120 - y) / 3000
```

```
    If x > 120 And x < 5120 And y > 120 And y < 3120 Then
```

```
        Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
```

```
    End If
```

```
End Sub
```

```
Private Sub cmdacceptarinput3_Click()
```

```
    For lazo = 0 To Lstterminos.ListCount - 1
```

```
        frmruleblock1.Combo1(3).List(lazo) = Lstterminos.List(lazo)
```

```
    Next lazo
```

```
    frminput3.Hide
```

```
End Sub
```

```
Private Sub Lstterminos_Click()
```

```
    If Lstterminos.ListIndex = 0 Then
```

```
        For lazo = 1 To 4
```

```
            If Line17(lazo).Visible = True Then
```

```
                Line17(lazo).BorderWidth = 2
```

```
            End If
```

```
        Next lazo
```

```
        For lazo = 5 To 20
```

```
            If Line17(lazo).Visible = True Then
```

```

        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 1 Then
    For lazo = 1 To 4
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 5 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 9 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 2 Then
    For lazo = 1 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 9 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 13 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo

```

```

End If
If Lstterminos.ListIndex = 3 Then
    For lazo = 1 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 13 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
End If
End Sub

```

```

Private Sub Lstterminos_DblClick()

```

```

    If Lstterminos.ListIndex = 0 Then
        frmtermino1prop3.Show
    End If

```

```

If Lstterminos.ListIndex = 1 Then
    frmtermino2prop3.Show
End If

```

```

If Lstterminos.ListIndex = 2 Then
    frmtermino3prop3.Show
End If

```

```

If Lstterminos.ListIndex = 3 Then
    frmtermino4prop3.Show
End If

```

```

If Lstterminos.ListIndex = 4 Then
    frmtermino5prop3.Show
End If

```

```
End Sub
```

```
Private Sub tbrdefin3_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Key
```

```
Case "addterm"
```

```
    matriz_indices(3, 2) = matriz_indices(3, 2) + 1
```

```
    If Lstterminos.ListCount = 3 Then
```

```
        Lstterminos.AddItem "term" + Str(4)
```

```
        frmruleblock1.Combo1(3).AddItem "term" + Str(4)
```

```
        frmtermino4prop3.Show
```

```
    Else
```

```
        If Lstterminos.ListCount = 4 Then
```

```
            Lstterminos.AddItem "term" + Str(5)
```

```
            frmruleblock1.Combo1(3).AddItem "term" + Str(5)
```

```
            frmtermino5prop3.Show
```

```
        Else
```

```
            matriz_indices(3, 2) = 5
```

```
            MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"
```

```
        End If
```

```
    End If
```

```
End Select
```

End Sub

Private Sub Text1_Change()

VarNames(3) = Text1.Text

frminput3.Caption = "Definición de Entrada " + VarNames(3)

End Sub

Private Sub Text2_Change()

xmin = Val(Text2.Text)

Universo_entrada(3, 1) = xmin

gap = (xmax - xmin) / 5

For lazo = 0 To 5

 Text4(lazo + 1) = Str(xmin + gap * lazo)

Next lazo

frmtermino1prop3.Text5.Text = Universo_entrada(3, 1)

End Sub

Private Sub Text3_Change()

xmax = Val(Text3.Text)

Universo_entrada(3, 2) = xmax

gap = (xmax - xmin) / 5

For lazo = 0 To 5

 Text4(lazo + 1) = Str(xmin + gap * lazo)

Next lazo

frmtermino3prop3.Text8.Text = Universo_entrada(3, 2)

End Sub

Dim xpos, ypos As Double

Dim xmin, xmax, gap As Double

Private Sub Form_Activate()

Call Text2_Change

Call Text3_Change

End Sub

Private Sub Form_Click()

```

For lazo = 1 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End Sub

```

```

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
ypos = (3120 - y) / 3000
If x > 120 And x < 5120 And y > 120 And y < 3120 Then
    Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
End If
End Sub

```

```

Private Sub cmdacceptarinput4_Click()
For lazo = 0 To Lstterminos.ListCount - 1
    frmruleblock1.Combo1(4).List(lazo) = Lstterminos.List(lazo)
Next lazo
frminput4.Hide
End Sub

```

```

Private Sub Lstterminos_Click()
If Lstterminos.ListIndex = 0 Then
    For lazo = 1 To 4
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 5 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 1 Then
    For lazo = 1 To 4

```



```

    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
For lazo = 5 To 8
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 2
    End If
Next lazo
For lazo = 9 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 2 Then
    For lazo = 1 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 9 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 13 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 3 Then
    For lazo = 1 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    
```

```

Next lazo
For lazo = 13 To 16
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 2
    End If
Next lazo
For lazo = 17 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
End If
End Sub

```

```

Private Sub Lstterminos_DblClick()

```

```

    If Lstterminos.ListIndex = 0 Then
        frmtermino1prop4.Show
    End If

```

```

    If Lstterminos.ListIndex = 1 Then
        frmtermino2prop4.Show
    End If

```

```

    If Lstterminos.ListIndex = 2 Then
        frmtermino3prop4.Show
    End If

```

End If

If Lstterminos.ListIndex = 3 Then

frmtermino4prop4.Show

End If

If Lstterminos.ListIndex = 4 Then

frmtermino5prop4.Show

End If

End Sub

Private Sub tbrdefin4_ButtonClick(ByVal Button As MSComctlLib.Button)

Select Case Button.Key

Case "addterm"

matriz_indices(4, 2) = matriz_indices(4, 2) + 1

If Lstterminos.ListCount = 3 Then

Lstterminos.AddItem "term" + Str(4)

frmruleblock1.Combo1(4).AddItem "term" + Str(4)

frmtermino4prop4.Show

Else

If Lstterminos.ListCount = 4 Then

Lstterminos.AddItem "term" + Str(5)

frmruleblock1.Combo1(4).AddItem "term" + Str(5)

frmtermino5prop4.Show

Else

matriz_indices(4, 2) = 5

MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"

End If

End If

End Select

End Sub

Private Sub Text1_Change()

VarNames(4) = Text1.Text

frminput4.Caption = "Definición de Entrada " + VarNames(4)

End Sub

```
Private Sub Text2_Change()
```

```
xmin = Val(Text2.Text)
```

```
Universo_entrada(4, 1) = xmin
```

```
gap = (xmax - xmin) / 5
```

```
For lazo = 0 To 5
```

```
    Text4(lazo + 1) = Str(xmin + gap * lazo)
```

```
Next lazo
```

```
frmtermino1prop4.Text5.Text = Universo_entrada(4, 1)
```

```
End Sub
```

```
Private Sub Text3_Change()
```

```
xmax = Val(Text3.Text)
```

```
Universo_entrada(4, 2) = xmax
```

```
gap = (xmax - xmin) / 5
```

```
For lazo = 0 To 5
```

```
    Text4(lazo + 1) = Str(xmin + gap * lazo)
```

```
Next lazo
```

```
frmtermino3prop4.Text8.Text = Universo_entrada(4, 2)
```

```
End Sub
```

```
Dim xpos, ypos As Double
```

```
Dim xmin, xmax, gap As Double
```

```
Private Sub Form_Activate()
```

```
Call Text2_Change
```

```
Call Text3_Change
```

```
End Sub
```

```
Private Sub Form_Click()
```

```
For lazo = 1 To 20
```

```
    If Line17(lazo).Visible = True Then
```

```
        Line17(lazo).BorderWidth = 1
```

```
    End If
```

```
Next lazo
```

```
End Sub
```

```
Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y5 As Single)
```

```
xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
```

```
ypos = (3120 - y) / 3000
```

```
If x > 120 And x < 5120 And y > 120 And y < 3120 Then
```

```
    Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
```

```
End If
```

```
End Sub
```

```
Private Sub cmdacceptarinput5_Click()
```

```
For lazo = 0 To Lstterminos.ListCount - 1
```

```
    frmruleblock1.Combo1(5).List(lazo) = Lstterminos.List(lazo)
```

```
Next lazo
```

```
frminput5.Hide
```

```
End Sub
```

```
Private Sub Lstterminos_Click()
```

```
If Lstterminos.ListIndex = 0 Then
```

```
    For lazo = 1 To 4
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 2
```

```
        End If
```

```
    Next lazo
```

```
    For lazo = 5 To 20
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 1
```

```
        End If
```

```
    Next lazo
```

```
End If
```

```
If Lstterminos.ListIndex = 1 Then
```

```
    For lazo = 1 To 4
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 1
```

```
        End If
```

```
    Next lazo
```

```
For lazo = 5 To 8
```

```
    If Line17(lazo).Visible = True Then
```

```
        Line17(lazo).BorderWidth = 2
```

```

    End If
Next lazo
For lazo = 9 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 2 Then
    For lazo = 1 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 9 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 13 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 3 Then
    For lazo = 1 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 13 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 17 To 20

```

```

        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
End If
End Sub

```

```

Private Sub Lstterminos_DblClick()

```

```

    If Lstterminos.ListIndex = 0 Then
        frmtermino1prop5.Show
    End If

```

```

    If Lstterminos.ListIndex = 1 Then
        frmtermino2prop5.Show
    End If

```

```

    If Lstterminos.ListIndex = 2 Then
        frmtermino3prop5.Show
    End If

```

```

    If Lstterminos.ListIndex = 3 Then
        frmtermino4prop5.Show
    End If

```

```

    If Lstterminos.ListIndex = 4 Then

```

```

    frmtermino5prop5.Show
End If

```

```

End Sub

```

```

Private Sub tbrdefin5_ButtonClick(ByVal Button As MSComctlLib.Button)

```

```

    Select Case Button.Key

```

```

        Case "addterm"

```

```

            matriz_indices(5, 2) = matriz_indices(5, 2) + 1

```

```

            If Lstterminos.ListCount = 3 Then

```

```

                Lstterminos.AddItem "term" + Str(4)

```

```

                frmruleblock1.Combo1(5).AddItem "term" + Str(4)

```

```

                frmtermino4prop5.Show

```

```

            Else

```

```

                If Lstterminos.ListCount = 4 Then

```

```

                    Lstterminos.AddItem "term" + Str(5)

```

```

                    frmruleblock1.Combo1(5).AddItem "term" + Str(5)

```

```

                    frmtermino5prop5.Show

```

```

                Else

```

```

                    matriz_indices(5, 2) = 5

```

```

                    MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"

```

```

                End If

```

```

            End If

```

```

        End Select

```

```

    End Sub

```

```

Private Sub Text1_Change()

```

```

    VarNames(5) = Text1.Text

```

```

    frminput5.Caption = "Definición de Entrada " + VarNames(5)

```

```

End Sub

```

```

Private Sub Text2_Change()

```

```

    xmin = Val(Text2.Text)

```

```

    Universo_entrada(5, 1) = xmin

```

```

    gap = (xmax - xmin) / 5

```

```

    For lazo = 0 To 5

```

```

        Text4(lazo + 1) = Str(xmin + gap * lazo)
    
```



```

Next lazo
frmtermino1prop5.Text5.Text = Universo_entrada(5, 1)
End Sub

```

```

Private Sub Text3_Change()
xmax = Val(Text3.Text)
Universo_entrada(5, 2) = xmax
gap = (xmax - xmin) / 5
For lazo = 0 To 5
    Text4(lazo + 1) = Str(xmin + gap * lazo)
Next lazo
frmtermino3prop5.Text8.Text = Universo_entrada(5, 2)
End Sub

```

```

Dim xpos, ypos As Double
Dim xmin, xmax, gap As Double

```

```

Private Sub Form_Click()
For lazo = 1 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End Sub

```

```

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
ypos = (3120 - y) / 3000
If x > 120 And x < 5120 And y > 120 And y < 3120 Then
    Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
End If
End Sub

```

```

Private Sub cmdacceptarinput1_Click()
For lazo = 0 To Lstterminos.ListCount - 1
    frmruleblock1.Combo6(1).List(lazo) = Lstterminos.List(lazo)
Next lazo
frmoutput1.Hide

```

End Sub

Private Sub Form_Load()

Set tbrdefin10.ImageList = ImageList1

With tbrdefin10.Buttons

.Add 1, "addterm", , tbrDefault, "addterm"

.Item(1).ToolTipText = "agregar término"

End With

xmin = 0

xmax = 100

Text2.Text = Str(xmin)

Text3.Text = Str(xmax)

End Sub

Private Sub Form_Unload(Cancel As Integer)

frmoutput1.Hide

End Sub

Private Sub Lstterminos_Click()

If Lstterminos.ListIndex = 0 Then

For lazo = 1 To 4

If Line17(lazo).Visible = True Then

Line17(lazo).BorderWidth = 2

End If

Next lazo

For lazo = 5 To 20

If Line17(lazo).Visible = True Then

Line17(lazo).BorderWidth = 1

End If

Next lazo

End If

If Lstterminos.ListIndex = 1 Then

For lazo = 1 To 4

If Line17(lazo).Visible = True Then

Line17(lazo).BorderWidth = 1

End If

Next lazo

For lazo = 5 To 8

If Line17(lazo).Visible = True Then

```

        Line17(lazo).BorderWidth = 2
    End If
Next lazo
For lazo = 9 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 2 Then
    For lazo = 1 To 8
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 9 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 13 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 3 Then
    For lazo = 1 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 13 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo

```

```

For lazo = 17 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
End If
End Sub

```

```

Private Sub Lstterminos_DblClick()

```

```

    If Lstterminos.ListIndex = 0 Then
        frmoutput1prop1.Show
    End If

```

```

    If Lstterminos.ListIndex = 1 Then
        frmoutput1prop2.Show
    End If

```

```

    If Lstterminos.ListIndex = 2 Then
        frmoutput1prop3.Show
    End If

```

```

    If Lstterminos.ListIndex = 3 Then
        frmoutput1prop4.Show
    End If

```

```
If Lstterminos.ListIndex = 4 Then
```

```
    frmoutput1prop5.Show
```

```
End If
```

```
End Sub
```

```
Private Sub tbrdefin10_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Select Case Button.Key
```

```
    Case "addterm"
```

```
        matriz_indices(1, 4) = matriz_indices(1, 4) + 1
```

```
        If Lstterminos.ListCount = 3 Then
```

```
            Lstterminos.AddItem "term" + Str(4)
```

```
            frmruleblock1.Combo6(1).AddItem "term" + Str(4)
```

```
            frmoutput1prop4.Show
```

```
        Else
```

```
            If Lstterminos.ListCount = 4 Then
```

```
                Lstterminos.AddItem "term" + Str(5)
```

```
                frmruleblock1.Combo6(1).AddItem "term" + Str(5)
```

```
                frmoutput1prop5.Show
```

```
            Else
```

```
                matriz_indices(1, 4) = 5
```

```
                MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"
```

```
            End If
```

```
        End If
```

```
End Select
```

```
End Sub
```

```
Private Sub Text1_Change()
```

```
    OutNames(1) = Text1.Text
```

```
    frmoutput1.Caption = "Definición de salida " + OutNames(1)
```

```
End Sub
```

```
Private Sub Text2_Change()
```

```
    xmin = Val(Text2.Text)
```

```
    Universo_salida(1, 1) = xmin
```

```
    gap = (xmax - xmin) / 5
```

```
    For lazo = 0 To 5
```

```
        Text4(lazo + 1) = Str(xmin + gap * lazo)
```

Next lazo

End Sub

Private Sub Text3_Change()

xmax = Val(Text3.Text)

Universe_salida(1, 2) = xmax

gap = (xmax - xmin) / 5

For lazo = 0 To 5

 Text4(lazo + 1) = Str(xmin + gap * lazo)

Next lazo

End Sub

Option Explicit

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()

frmoutput1prop1.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "0"

Text2.Text = "25"

Text3.Text = "50"

Text4.Text = "0"

Text9.Text = frmoutput1.Lstterminos.List(0)

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

frmoutput1.Lstterminos.List(0) = Text9.Text

If Text9.Text <> "term 1" Then

 frmoutput1prop1.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120

 If Text1.Text = "0" Then

```

    Y1 = 3120
Else
    If Text1.Text = "1" Then
        Y1 = 120
    Else
        MsgBox "La Ordenada para este punto solo puede tener valor de cero o uno", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If
X2 = 120 + CInt(Val(Text2.Text)) * 50
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * 50
Y4 = 3120
frmoutput1.Line17(7).BorderColor = &HFF
frmoutput1.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120
    If Text1.Text = "0" Then
        Y1 = 3120
    Else
        If Text1.Text = "1" Then
            Y1 = 120
        Else
            MsgBox "La Ordenada para este punto solo puede tener valor de cero o uno", vbExclamation,
            "Atención"
            GoTo ignoredatos
        End If
    End If
End If

```

```

    End If
End If
X2 = 120 + CInt(Val(Text2.Text)) * 50
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 3120
X4 = X3
Y4 = Y3
frmoutput1.Line17(7).BorderColor = &H0
frmoutput1.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If
With frmoutput1
    .Line17(1).X1 = X1
    .Line17(1).Y1 = Y1
    .Line17(1).X2 = X2
    .Line17(1).Y2 = Y2
    .Line17(1).Visible = True
    .Line17(2).X1 = X2
    .Line17(2).Y1 = Y2
    .Line17(2).X2 = X3
    .Line17(2).Y2 = Y3
    .Line17(2).Visible = True

```



```

.Line17(3).X1 = X3
.Line17(3).Y1 = Y3
.Line17(3).X2 = X4
.Line17(3).Y2 = Y4
.Line17(3).Visible = True
.Line17(4).X1 = X4
.Line17(4).Y1 = Y4
.Line17(4).X2 = 5120
.Line17(4).Y2 = 3120
.Line17(4).Visible = True
End With
frmoutput1.prop1.Hide
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text1.Locked = True
    Text7.Text = "0"
End If
End Sub

```

```

Private Sub Text9_Change()

```

```
frmoutput1prop1.Caption = "Propiedades del término " + Text9.Text
End Sub
```

```
Option Explicit
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Private Sub CancelButton_Click()
```

```
frmoutput1prop2.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "25"
```

```
Text2.Text = "50"
```

```
Text3.Text = "75"
```

```
Text4.Text = "0"
```

```
Text9.Text = frmoutput1.Lstterminos.List(1)
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
frmoutput1.Lstterminos.List(1) = Text9.Text
```

```
If Text9.Text <> "term 2" Then
```

```
    frmoutput1prop2.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * 50
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * 50
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * 50
```

```
    Y3 = 120
```

```
    X4 = 120 + CInt(Val(Text4.Text)) * 50
```

```
    Y4 = 3120
```

```
    frmoutput1.Line17(7).BorderColor = &HFF
```

```
    frmoutput1.Line17(8).BorderColor = &H0
```

```
    If X3 >= X4 Then
```

```

    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"

```

```

    GoTo ignoredatos

```

```

End If

```

```

End If

```

```

If opttriangular.Value = True Then

```

```

    X1 = 120 + CInt(Val(Text1.Text)) * 50

```

```

    Y1 = 3120

```

```

    X2 = 120 + CInt(Val(Text2.Text)) * 50

```

```

    Y2 = 120

```

```

    X3 = 120 + CInt(Val(Text3.Text)) * 50

```

```

    Y3 = 3120

```

```

    X4 = X3

```

```

    Y4 = Y3

```

```

    frmoutput1.Line17(7).BorderColor = &H0

```

```

    frmoutput1.Line17(8).BorderColor = &H0

```

```

End If

```

```

If X2 >= X3 Then

```

```

    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"

```

```

    GoTo ignoredatos

```

```

End If

```

```

If opttriangular.Value = True And X3 > 5120 Then

```

```

    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"

```

```

    GoTo ignoredatos

```

```

Else

```

```

    If Opttrapezoidal.Value = True And X4 > 5120 Then

```

```

        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"

```

```

        GoTo ignoredatos

```

```

    End If

```

```

End If

```

```

With frmoutput1

```

```

    .Line17(5).X1 = X1

```

```

    .Line17(5).Y1 = Y1

```

```

.Line17(5).X2 = X2
.Line17(5).Y2 = Y2
.Line17(5).Visible = True
.Line17(6).X1 = X2
.Line17(6).Y1 = Y2
.Line17(6).X2 = X3
.Line17(6).Y2 = Y3
.Line17(6).Visible = True
.Line17(7).X1 = X3
.Line17(7).Y1 = Y3
.Line17(7).X2 = X4
.Line17(7).Y2 = Y4
.Line17(7).Visible = True
.Line17(8).X1 = X4
.Line17(8).Y1 = Y4
.Line17(8).X2 = 5120
.Line17(8).Y2 = 3120
.Line17(8).Visible = True
End With
frmoutput1.prop2.Hide
ignoredatos:
End Sub

```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
Label4.Enabled = True
```

```
Text4.Enabled = True
```

```
Text8.Enabled = True
```

```
Label8.Enabled = True
```

```
Text7.Text = "1"
```

```
Text1.Locked = False
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
Label4.Enabled = False
```

```

Text4.Enabled = False
Text8.Enabled = False
Label8.Enabled = False
Text7.Text = "0"
End If
End Sub
Private Sub Text9_Change()
frmoutput1prop2.Caption = "Propiedades del término " + Text9.Text
End Sub

Option Explicit
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()
frmoutput1prop3.Hide
End Sub

Private Sub Form_Load()
Text1.Text = "50"
Text2.Text = "75"
Text3.Text = "100"
Text4.Text = "1"
Text9.Text = frmoutput1.Lstterminos.List(2)
opttriangular.Value = True
Opttrapezoidal.Value = False
End Sub

Private Sub OKButton_Click()
frmoutput1.Lstterminos.List(2) = Text9.Text
If Text9.Text <> "term 3" Then
    frmoutput1prop3.Caption = "Propiedades de " + Text9.Text
End If
If Opttrapezoidal.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120

```

```

X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 120
X4 = 5120
If Text4.Text = "0" Then
    Y4 = 3120
    frmoutput1.Line17(11).BorderColor = &HC000&
    frmoutput1.Line17(11).BorderStyle = 1
Else
    If Text4.Text = "1" Then
        Y4 = 120
        frmoutput1.Line17(11).BorderColor = &H0&
        frmoutput1.Line17(11).BorderStyle = 3
    Else
        MsgBox "La ordenada para el punto 3 solo puede tomar valor de cero o uno", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If
If X3 > X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If
If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frmoutput1.Line17(12).BorderColor = &H0
End If

If X2 >= X3 Then

```

```
MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
"Atención"
```

```
GoTo ignoredatos
```

```
End If
```

```
With frmoutput1
```

```
.Line17(9).X1 = X1
```

```
.Line17(9).Y1 = Y1
```

```
.Line17(9).X2 = X2
```

```
.Line17(9).Y2 = Y2
```

```
.Line17(9).Visible = True
```

```
.Line17(10).X1 = X2
```

```
.Line17(10).Y1 = Y2
```

```
.Line17(10).X2 = X3
```

```
.Line17(10).Y2 = Y3
```

```
.Line17(10).Visible = True
```

```
.Line17(11).X1 = X3
```

```
.Line17(11).Y1 = Y3
```

```
.Line17(11).X2 = X4
```

```
.Line17(11).Y2 = Y4
```

```
.Line17(11).Visible = True
```

```
.Line17(12).X1 = X4
```

```
.Line17(12).Y1 = Y4
```

```
.Line17(12).X2 = 5120
```

```
.Line17(12).Y2 = 3120
```

```
.Line17(12).Visible = True
```

```
End With
```

```
frmoutput1prop3.Hide
```

```
ignoredatos:
```

```
End Sub
```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
Text3.Locked = False
```

```
Text7.Text = "1"
```

```
Label4.Enabled = True
```

```
Text4.Enabled = True
```

```
Text8.Enabled = True
```

```
Label8.Enabled = True
Text4.Text = 1
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
Text3.Locked = True
```

```
Text7.Text = "0"
```

```
Label4.Enabled = False
```

```
Text4.Enabled = False
```

```
Text8.Enabled = False
```

```
Label8.Enabled = False
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmoutput1prop3.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Option Explicit
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Private Sub CancelButton_Click()
```

```
frmoutput1prop4.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "12.5"
```

```
Text2.Text = "25"
```

```
Text3.Text = "37.5"
```

```
Text4.Text = "0"
```

```
Text9.Text = frmoutput1.Lstterminos.List(3)
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```



```
frmoutput1.Lstterminos.List(3) = Text9.Text
```

```
If Text9.Text <> "term 4" Then
```

```
    frmoutput1.prop4.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * 50
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * 50
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * 50
```

```
    Y3 = 120
```

```
    X4 = 120 + CInt(Val(Text4.Text)) * 50
```

```
    Y4 = 3120
```

```
    frmoutput1.Line17(7).BorderColor = &HFF
```

```
    frmoutput1.Line17(8).BorderColor = &H0
```

```
    If X3 >= X4 Then
```

```
        MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,  
        "Atención"
```

```
        GoTo ignoredatos
```

```
    End If
```

```
End If
```

```
If opttriangular.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * 50
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * 50
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * 50
```

```
    Y3 = 3120
```

```
    X4 = X3
```

```
    Y4 = Y3
```

```
    frmoutput1.Line17(7).BorderColor = &H0
```

```
    frmoutput1.Line17(8).BorderColor = &H0
```

```
End If
```

```
If X2 >= X3 Then
```

```

    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If
With frmoutput1
    .Line17(13).X1 = X1
    .Line17(13).Y1 = Y1
    .Line17(13).X2 = X2
    .Line17(13).Y2 = Y2
    .Line17(13).Visible = True
    .Line17(14).X1 = X2
    .Line17(14).Y1 = Y2
    .Line17(14).X2 = X3
    .Line17(14).Y2 = Y3
    .Line17(14).Visible = True
    .Line17(15).X1 = X3
    .Line17(15).Y1 = Y3
    .Line17(15).X2 = X4
    .Line17(15).Y2 = Y4
    .Line17(15).Visible = True
    .Line17(16).X1 = X4
    .Line17(16).Y1 = Y4
    .Line17(16).X2 = 5120
    .Line17(16).Y2 = 3120
    .Line17(16).Visible = True
End With
frmoutput1prop4.Hide
ignoredatos:

```

End Sub

Private Sub Opttrapezoidal_Click()

If Opttrapezoidal.Value = True Then

Label4.Enabled = True

Text4.Enabled = True

Text8.Enabled = True

Label8.Enabled = True

Text7.Text = "1"

Text1.Locked = False

End If

End Sub

Private Sub opttriangular_Click()

If opttriangular.Value = True Then

Label4.Enabled = False

Text4.Enabled = False

Text8.Enabled = False

Label8.Enabled = False

Text7.Text = "0"

End If

End Sub

Private Sub Text9_Change()

frmoutput1prop4.Caption = "Propiedades del término " + Text9.Text

End Sub

Option Explicit

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()

frmoutput1prop5.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "62.5"

Text2.Text = "75"

Text3.Text = "87.5"

```

Text4.Text = "0"
Text9.Text = frmoutput1.Lstterminos.List(4)
opttriangular.Value = True
Opttrapezoidal.Value = False
End Sub

```

Private Sub OKButton_Click()

```

frmoutput1.Lstterminos.List(4) = Text9.Text
If Text9.Text <> "term 5" Then
    frmoutput1.prop5.Caption = "Propiedades de " + Text9.Text
End If
If Opttrapezoidal.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 120
    X4 = 120 + CInt(Val(Text4.Text)) * 50
    Y4 = 3120
    frmoutput1.Line17(7).BorderColor = &HFF
    frmoutput1.Line17(8).BorderColor = &H0
    If X3 >= X4 Then
        MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 3120

```

```

X4 = X3
Y4 = Y3
frmoutput1.Line17(19).BorderColor = &H0
frmoutput1.Line17(20).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

With frmoutput1
    .Line17(17).X1 = X1
    .Line17(17).Y1 = Y1
    .Line17(17).X2 = X2
    .Line17(17).Y2 = Y2
    .Line17(17).Visible = True
    .Line17(18).X1 = X2
    .Line17(18).Y1 = Y2
    .Line17(18).X2 = X3
    .Line17(18).Y2 = Y3
    .Line17(18).Visible = True
    .Line17(19).X1 = X3
    .Line17(19).Y1 = Y3
    .Line17(19).X2 = X4
    .Line17(19).Y2 = Y4
    .Line17(19).Visible = True
    .Line17(20).X1 = X4

```

```
.Line17(20).Y1 = Y4
.Line17(20).X2 = 5120
.Line17(20).Y2 = 3120
.Line17(20).Visible = True
```

```
End With
```

```
frmoutput1prop5.Hide
```

```
ignoredatos:
```

```
End Sub
```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text1.Locked = False
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmoutput1prop5.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Dim xpos, ypos As Double
```

```
Dim xmin, xmax, gap As Double
```

```
Private Sub Form_Click()
```

```
For lazo = 1 To 20
```

```

    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End Sub

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
    xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
    ypos = (3120 - y) / 3000
    If x > 120 And x < 5120 And y > 120 And y < 3120 Then
        Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
    End If
End Sub

Private Sub Form_Load()
    Set tbrdefin11.ImageList = ImageList1
    With tbrdefin11.Buttons
        .Add 1, "addterm", , tbrDefault, "addterm"
        .Item(1).ToolTipText = "agregar término"
    End With
    xmin = 0
    xmax = 100
    Text2.Text = Str(xmin)
    Text3.Text = Str(xmax)
End Sub

Private Sub Lstterminos_Click()
    If Lstterminos.ListIndex = 0 Then
        For lazo = 1 To 4
            If Line17(lazo).Visible = True Then
                Line17(lazo).BorderWidth = 2
            End If
        Next lazo
        For lazo = 5 To 20
            If Line17(lazo).Visible = True Then
                Line17(lazo).BorderWidth = 1
            End If
        Next lazo
    End If
End Sub

```

```

If Lstterminos.ListIndex = 1 Then
  For lazo = 1 To 4
    If Line17(lazo).Visible = True Then
      Line17(lazo).BorderWidth = 1
    End If
  Next lazo
  For lazo = 5 To 8
    If Line17(lazo).Visible = True Then
      Line17(lazo).BorderWidth = 2
    End If
  Next lazo
  For lazo = 9 To 20
    If Line17(lazo).Visible = True Then
      Line17(lazo).BorderWidth = 1
    End If
  Next lazo
End If
If Lstterminos.ListIndex = 2 Then
  For lazo = 1 To 8
    If Line17(lazo).Visible = True Then
      Line17(lazo).BorderWidth = 1
    End If
  Next lazo
  For lazo = 9 To 12
    If Line17(lazo).Visible = True Then
      Line17(lazo).BorderWidth = 2
    End If
  Next lazo
  For lazo = 13 To 20
    If Line17(lazo).Visible = True Then
      Line17(lazo).BorderWidth = 1
    End If
  Next lazo
End If
If Lstterminos.ListIndex = 3 Then
  For lazo = 1 To 12
    If Line17(lazo).Visible = True Then

```



```

        Line17(lazo).BorderWidth = 1
    End If
Next lazo
For lazo = 13 To 16
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 2
    End If
Next lazo
For lazo = 17 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
End If
End Sub

```

```

Private Sub Lstterminos_DblClick()

```

```

    If Lstterminos.ListIndex = 0 Then
        frmoutput2prop1.Show
    End If

```

```

    If Lstterminos.ListIndex = 1 Then
        frmoutput2prop2.Show
    End If

```

```

If Lstterminos.ListIndex = 2 Then
    frmoutput2prop3.Show
End If

```

```

If Lstterminos.ListIndex = 3 Then
    frmoutput2prop4.Show
End If

```

```

If Lstterminos.ListIndex = 4 Then
    frmoutput2prop5.Show
End If
End Sub

```

```

Private Sub cmdacceptarinput1_Click()
    frmoutput2.Hide
End Sub

```

```

Private Sub tbrdefin11_ButtonClick(ByVal Button As MSComctlLib.Button)
    Select Case Button.Key
        Case "addterm"
            matriz_indices(2, 4) = matriz_indices(2, 4) + 1
            If Lstterminos.ListCount = 3 Then
                Lstterminos.AddItem "term" + Str(4)
                frmruleblock1.Combo6(2).AddItem "term" + Str(4)
                frmoutput2prop4.Show
            Else
                If Lstterminos.ListCount = 4 Then
                    Lstterminos.AddItem "term" + Str(5)
                    frmruleblock1.Combo6(2).AddItem "term" + Str(5)
                    frmoutput2prop5.Show
                Else
                    matriz_indices(2, 4) = 5
                    MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"
                End If
            End If
        End If
    End Select

```

End Sub

Private Sub Text1_Change()

OutNames(2) = Text1.Text

frmoutput2.Caption = "Definición de salida " + OutNames(2)

End Sub

Private Sub Text2_Change()

xmin = Val(Text2.Text)

Universo_salida(2, 1) = xmin

gap = (xmax - xmin) / 5

For lazo = 0 To 5

 Text4(lazo + 1) = Str(xmin + gap * lazo)

Next lazo

End Sub

Private Sub Text3_Change()

xmax = Val(Text3.Text)

Universo_salida(2, 2) = xmax

gap = (xmax - xmin) / 5

For lazo = 0 To 5

 Text4(lazo + 1) = Str(xmin + gap * lazo)

Next lazo

End Sub

Option Explicit

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()

frmoutput2prop1.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "0"

Text2.Text = "25"

Text3.Text = "50"

Text4.Text = "0"

```
Text9.Text = frmoutput2.Lstterminos.List(0)
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
frmoutput2.Lstterminos.List(0) = Text9.Text
```

```
If Text9.Text <> "term 1" Then
```

```
    frmoutput2prop1.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120
```

```
    If Text1.Text = "0" Then
```

```
        Y1 = 3120
```

```
    Else
```

```
        If Text1.Text = "1" Then
```

```
            Y1 = 120
```

```
        Else
```

```
            MsgBox "La Ordenada para este punto solo puede tener valor de cero o uno", vbExclamation,  
"Atención"
```

```
            GoTo ignoredatos
```

```
        End If
```

```
    End If
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * 50
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * 50
```

```
    Y3 = 120
```

```
    X4 = 120 + CInt(Val(Text4.Text)) * 50
```

```
    Y4 = 3120
```

```
frmoutput2.Line17(7).BorderColor = &HFF
```

```
frmoutput2.Line17(8).BorderColor = &H0
```

```
If X3 >= X4 Then
```

```
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,  
"Atención"
```

```
    GoTo ignoredatos
```

```
End If
```

```
End If
```

```

If opttriangular.Value = True Then
    X1 = 120
    If Text1.Text = "0" Then
        Y1 = 3120
    Else
        If Text1.Text = "1" Then
            Y1 = 120
        Else
            MsgBox "La Ordenada para este punto solo puede tener valor de cero o uno", vbExclamation,
            "Atención"
            GoTo ignoredatos
        End If
    End If
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frmoutput2.Line17(7).BorderColor = &H0
    frmoutput2.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    
```

```

    End If
End If
With frmoutput2
    .Line17(1).X1 = X1
    .Line17(1).Y1 = Y1
    .Line17(1).X2 = X2
    .Line17(1).Y2 = Y2
    .Line17(1).Visible = True
    .Line17(2).X1 = X2
    .Line17(2).Y1 = Y2
    .Line17(2).X2 = X3
    .Line17(2).Y2 = Y3
    .Line17(2).Visible = True
    .Line17(3).X1 = X3
    .Line17(3).Y1 = Y3
    .Line17(3).X2 = X4
    .Line17(3).Y2 = Y4
    .Line17(3).Visible = True
    .Line17(4).X1 = X4
    .Line17(4).Y1 = Y4
    .Line17(4).X2 = 5120
    .Line17(4).Y2 = 3120
    .Line17(4).Visible = True
End With
frmoutput2prop1.Hide
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()

```

```

If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Locked = False
End If

```

End Sub

Private Sub opttriangular_Click()

If opttriangular.Value = True Then

Label4.Enabled = False

Text4.Enabled = False

Text8.Enabled = False

Label8.Enabled = False

Text1.Locked = True

Text7.Text = "0"

End If

End Sub

Private Sub Text9_Change()

frmoutput2prop1.Caption = "Propiedades del término " + Text9.Text

End Sub

Option Explicit

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()

frmoutput2prop2.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "25"

Text2.Text = "50"

Text3.Text = "75"

Text4.Text = "0"

Text9.Text = frmoutput2.Lstterminos.List(1)

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

frmoutput2.Lstterminos.List(1) = Text9.Text

If Text9.Text <> "term 2" Then

frmoutput2prop2.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

X1 = 120 + CInt(Val(Text1.Text)) * 50

Y1 = 3120

X2 = 120 + CInt(Val(Text2.Text)) * 50

Y2 = 120

X3 = 120 + CInt(Val(Text3.Text)) * 50

Y3 = 120

X4 = 120 + CInt(Val(Text4.Text)) * 50

Y4 = 3120

frmoutput2.Line17(7).BorderColor = &HFF

frmoutput2.Line17(8).BorderColor = &H0

If X3 >= X4 Then

MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
"Atención"

GoTo ignoredatos

End If

End If

If opttriangular.Value = True Then

X1 = 120 + CInt(Val(Text1.Text)) * 50

Y1 = 3120

X2 = 120 + CInt(Val(Text2.Text)) * 50

Y2 = 120

X3 = 120 + CInt(Val(Text3.Text)) * 50

Y3 = 3120

X4 = X3

Y4 = Y3

frmoutput2.Line17(7).BorderColor = &H0

frmoutput2.Line17(8).BorderColor = &H0

End If

If X2 >= X3 Then

MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
"Atención"

GoTo ignoredatos


```

End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If
With frmoutput2
    .Line17(5).X1 = X1
    .Line17(5).Y1 = Y1
    .Line17(5).X2 = X2
    .Line17(5).Y2 = Y2
    .Line17(5).Visible = True
    .Line17(6).X1 = X2
    .Line17(6).Y1 = Y2
    .Line17(6).X2 = X3
    .Line17(6).Y2 = Y3
    .Line17(6).Visible = True
    .Line17(7).X1 = X3
    .Line17(7).Y1 = Y3
    .Line17(7).X2 = X4
    .Line17(7).Y2 = Y4
    .Line17(7).Visible = True
    .Line17(8).X1 = X4
    .Line17(8).Y1 = Y4
    .Line17(8).X2 = 5120
    .Line17(8).Y2 = 3120
    .Line17(8).Visible = True
End With
frmoutput2prop2.Hide
ignoredatos:
End Sub

Private Sub Opttrapezoidal_Click()

```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text1.Locked = False
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmoutput2prop2.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Option Explicit
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Private Sub CancelButton_Click()
```

```
frmoutput2prop3.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "50"
```

```
Text2.Text = "75"
```

```
Text3.Text = "100"
```

```
Text4.Text = "1"
```

```
Text9.Text = frmoutput2.Lstterminos.List(2)
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
frmoutput2.Lstterminos.List(2) = Text9.Text
```

```
If Text9.Text <> "term 3" Then
```

```
    frmoutput2.prop3.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * 50
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * 50
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * 50
```

```
    Y3 = 120
```

```
    X4 = 5120
```

```
If Text4.Text = "0" Then
```

```
    Y4 = 3120
```

```
    frmoutput2.Line17(11).BorderColor = &HC000&
```

```
    frmoutput2.Line17(11).BorderStyle = 1
```

```
Else
```

```
    If Text4.Text = "1" Then
```

```
        Y4 = 120
```

```
        frmoutput2.Line17(11).BorderColor = &H0&
```

```
        frmoutput2.Line17(11).BorderStyle = 3
```

```
    Else
```

```
        MsgBox "La ordenada para el punto 3 solo puede tomar valor de cero o uno", vbExclamation,  
        "Atención"
```

```
        GoTo ignoredatos
```

```
    End If
```

```
End If
```

```
If X3 > X4 Then
```

```
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,  
    "Atención"
```

```
    GoTo ignoredatos
```

```
End If
```

```
End If
```

If opttriangular.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * 50

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * 50

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * 50

 Y3 = 3120

 X4 = X3

 Y4 = Y3

 frmoutput2.Line17(12).BorderColor = &H0

End If

If X2 >= X3 Then

 MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
 "Atención"

 GoTo ignoredatos

End If

With frmoutput2

 .Line17(9).X1 = X1

 .Line17(9).Y1 = Y1

 .Line17(9).X2 = X2

 .Line17(9).Y2 = Y2

 .Line17(9).Visible = True

 .Line17(10).X1 = X2

 .Line17(10).Y1 = Y2

 .Line17(10).X2 = X3

 .Line17(10).Y2 = Y3

 .Line17(10).Visible = True

 .Line17(11).X1 = X3

 .Line17(11).Y1 = Y3

 .Line17(11).X2 = X4

 .Line17(11).Y2 = Y4

 .Line17(11).Visible = True

 .Line17(12).X1 = X4

 .Line17(12).Y1 = Y4

 .Line17(12).X2 = 5120

 .Line17(12).Y2 = 3120

```

        .Line17(12).Visible = True
    End With
    frmoutput2prop3.Hide
    ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
    If Opttrapezoidal.Value = True Then
        Text3.Locked = False
        Text7.Text = "1"
        Label4.Enabled = True
        Text4.Enabled = True
        Text8.Enabled = True
        Label8.Enabled = True
        Text4.Text = 1
    End If
End Sub

```

```

Private Sub opttriangular_Click()
    If opttriangular.Value = True Then
        Text3.Locked = True
        Text7.Text = "0"
        Label4.Enabled = False
        Text4.Enabled = False
        Text8.Enabled = False
        Label8.Enabled = False
    End If
End Sub

```

```

Private Sub Text9_Change()
    frmoutput2prop3.Caption = "Propiedades del término " + Text9.Text
End Sub

```

```

Option Explicit
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

```

```

Private Sub CancelButton_Click()
    frmoutput2prop4.Hide

```

End Sub

Private Sub Form_Load()

Text1.Text = "12.5"

Text2.Text = "25"

Text3.Text = "37.5"

Text4.Text = "0"

Text9.Text = frmoutput2.Lstterminos.List(3)

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

frmoutput2.Lstterminos.List(3) = Text9.Text

If Text9.Text <> "term 4" Then

 frmoutput2prop4.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * 50

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * 50

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * 50

 Y3 = 120

 X4 = 120 + CInt(Val(Text4.Text)) * 50

 Y4 = 3120

 frmoutput2.Line17(7).BorderColor = &HFF

 frmoutput2.Line17(8).BorderColor = &H0

 If X3 >= X4 Then

 MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
 "Atención"

 GoTo ignoredatos

 End If

End If

If opttriangular.Value = True Then

```

X1 = 120 + CInt(Val(Text1.Text)) * 50
Y1 = 3120
X2 = 120 + CInt(Val(Text2.Text)) * 50
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 3120
X4 = X3
Y4 = Y3
frmoutput2.Line17(7).BorderColor = &H0
frmoutput2.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

With frmoutput2
    .Line17(13).X1 = X1
    .Line17(13).Y1 = Y1
    .Line17(13).X2 = X2
    .Line17(13).Y2 = Y2
    .Line17(13).Visible = True
    .Line17(14).X1 = X2
    .Line17(14).Y1 = Y2
    .Line17(14).X2 = X3
    .Line17(14).Y2 = Y3
    .Line17(14).Visible = True

```

```

.Line17(15).X1 = X3
.Line17(15).Y1 = Y3
.Line17(15).X2 = X4
.Line17(15).Y2 = Y4
.Line17(15).Visible = True
.Line17(16).X1 = X4
.Line17(16).Y1 = Y4
.Line17(16).X2 = 5120
.Line17(16).Y2 = 3120
.Line17(16).Visible = True

```

```
End With
```

```
frmoutput2prop4.Hide
```

```
ignoredatos:
```

```
End Sub
```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text1.Locked = False
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmoutput2prop4.Caption = "Propiedades del término " + Text9.Text
```


End Sub

Option Explicit

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()

frmoutput2prop5.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "62.5"

Text2.Text = "75"

Text3.Text = "87.5"

Text4.Text = "0"

Text9.Text = frmoutput2.Lstterminos.List(4)

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

frmoutput2.Lstterminos.List(4) = Text9.Text

If Text9.Text <> "term 5" Then

 frmoutput2prop5.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * 50

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * 50

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * 50

 Y3 = 120

 X4 = 120 + CInt(Val(Text4.Text)) * 50

 Y4 = 3120

 frmoutput2.Line17(7).BorderColor = &HFF

 frmoutput2.Line17(8).BorderColor = &H0

 If X3 >= X4 Then

```

    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"

```

```

    GoTo ignoredatos

```

```

End If

```

```

End If

```

```

If opttriangular.Value = True Then

```

```

    X1 = 120 + CInt(Val(Text1.Text)) * 50

```

```

    Y1 = 3120

```

```

    X2 = 120 + CInt(Val(Text2.Text)) * 50

```

```

    Y2 = 120

```

```

    X3 = 120 + CInt(Val(Text3.Text)) * 50

```

```

    Y3 = 3120

```

```

    X4 = X3

```

```

    Y4 = Y3

```

```

    frmoutput2.Line17(7).BorderColor = &H0

```

```

    frmoutput2.Line17(8).BorderColor = &H0

```

```

End If

```

```

If X2 >= X3 Then

```

```

    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"

```

```

    GoTo ignoredatos

```

```

End If

```

```

If opttriangular.Value = True And X3 > 5120 Then

```

```

    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"

```

```

    GoTo ignoredatos

```

```

Else

```

```

    If Opttrapezoidal.Value = True And X4 > 5120 Then

```

```

        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"

```

```

        GoTo ignoredatos

```

```

    End If

```

```

End If

```

```

With frmoutput2

```

```

    .Line17(17).X1 = X1

```

```

    .Line17(17).Y1 = Y1

```

```

.Line17(17).X2 = X2
.Line17(17).Y2 = Y2
.Line17(17).Visible = True
.Line17(18).X1 = X2
.Line17(18).Y1 = Y2
.Line17(18).X2 = X3
.Line17(18).Y2 = Y3
.Line17(18).Visible = True
.Line17(19).X1 = X3
.Line17(19).Y1 = Y3
.Line17(19).X2 = X4
.Line17(19).Y2 = Y4
.Line17(19).Visible = True
.Line17(20).X1 = X4
.Line17(20).Y1 = Y4
.Line17(20).X2 = 5120
.Line17(20).Y2 = 3120
.Line17(20).Visible = True
End With
frmoutput2prop5.Hide
ignoredatos:
End Sub

```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text1.Locked = False
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```

Text4.Enabled = False
Text8.Enabled = False
Label8.Enabled = False
Text7.Text = "0"
End If
End Sub
Private Sub Text9_Change()
frmoutput2prop5.Caption = "Propiedades del término " + Text9.Text
End Sub

```

```

Dim xpos, ypos As Double
Dim xmin, xmax, gap As Double

```

```

Private Sub Form_Click()
For lazo = 1 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End Sub

```

```

Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, x As Single, y As Single)
xpos = xmin + (x - 120) * ((xmax - xmin) / 5000)
ypos = (3120 - y) / 3000
If x > 120 And x < 5120 And y > 120 And y < 3120 Then
    Frame1.ToolTipText = Str(xpos) + "," + Str(ypos)
End If
End Sub

```

```

Private Sub Form_Load()
Set tbrdefin12.ImageList = ImageList1
With tbrdefin12.Buttons
    .Add 1, "addterm", , tbrDefault, "addterm"
    .Item(1).ToolTipText = "agregar término"
End With
xmin = 0
xmax = 100

```

```
Text2.Text = Str(xmin)
```

```
Text3.Text = Str(xmax)
```

```
End Sub
```

```
Private Sub Lstterminos_Click()
```

```
If Lstterminos.ListIndex = 0 Then
```

```
    For lazo = 1 To 4
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 2
```

```
        End If
```

```
    Next lazo
```

```
    For lazo = 5 To 20
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 1
```

```
        End If
```

```
    Next lazo
```

```
End If
```

```
If Lstterminos.ListIndex = 1 Then
```

```
    For lazo = 1 To 4
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 1
```

```
        End If
```

```
    Next lazo
```

```
    For lazo = 5 To 8
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 2
```

```
        End If
```

```
    Next lazo
```

```
    For lazo = 9 To 20
```

```
        If Line17(lazo).Visible = True Then
```

```
            Line17(lazo).BorderWidth = 1
```

```
        End If
```

```
    Next lazo
```

```
End If
```

```
If Lstterminos.ListIndex = 2 Then
```

```
    For lazo = 1 To 8
```

```
        If Line17(lazo).Visible = True Then
```

```

        Line17(lazo).BorderWidth = 1
    End If
Next lazo
For lazo = 9 To 12
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 2
    End If
Next lazo
For lazo = 13 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 1
    End If
Next lazo
End If
If Lstterminos.ListIndex = 3 Then
    For lazo = 1 To 12
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
    For lazo = 13 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 2
        End If
    Next lazo
    For lazo = 17 To 20
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo
End If
If Lstterminos.ListIndex = 4 Then
    For lazo = 1 To 16
        If Line17(lazo).Visible = True Then
            Line17(lazo).BorderWidth = 1
        End If
    Next lazo

```

```

For lazo = 17 To 20
    If Line17(lazo).Visible = True Then
        Line17(lazo).BorderWidth = 2
    End If
Next lazo
End If
End Sub
Private Sub Lstterminos_DblClick()
If Lstterminos.ListIndex = 0 Then
    frmoutput3prop1.Show
End If

If Lstterminos.ListIndex = 1 Then
    frmoutput3prop2.Show
End If

If Lstterminos.ListIndex = 2 Then
    frmoutput3prop3.Show
End If

If Lstterminos.ListIndex = 3 Then
    frmoutput3prop4.Show
End If

If Lstterminos.ListIndex = 4 Then
    frmoutput3prop5.Show
End If
End Sub

Private Sub cmdacceptarinput1_Click()
frmoutput3.Hide
End Sub

Private Sub tbrdefin12_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
    Case "addterm"
        matriz_indices(3, 4) = matriz_indices(3, 4) + 1

```

```

If Lstterminos.ListCount = 3 Then
    Lstterminos.AddItem "term" + Str(4)
    frmruleblock1.Combo6(3).AddItem "term" + Str(4)
    frmoutput3prop4.Show
Else
    If Lstterminos.ListCount = 4 Then
        Lstterminos.AddItem "term" + Str(5)
        frmruleblock1.Combo6(3).AddItem "term" + Str(5)
        frmoutput3prop5.Show
    Else
        matriz_indices(3, 4) = 5
        MsgBox "Se alcanzó el límite de términos", vbExclamation, "Atención"
    End If
End If
End Select
End Sub

```

```

Private Sub Text1_Change()
    OutNames(3) = Text1.Text
    frmoutput3.Caption = "Definición de salida " + OutNames(3)
End Sub

```

```

Private Sub Text2_Change()
    xmin = Val(Text2.Text)
    Universo_salida(3, 1) = xmin
    gap = (xmax - xmin) / 5
    For lazo = 0 To 5
        Text4(lazo + 1) = Str(xmin + gap * lazo)
    Next lazo
End Sub

```

```

Private Sub Text3_Change()
    xmax = Val(Text3.Text)
    Universo_salida(3, 2) = xmax
    gap = (xmax - xmin) / 5
    For lazo = 0 To 5
        Text4(lazo + 1) = Str(xmin + gap * lazo)
    Next lazo
End Sub

```


Next lazo

End Sub

Option Explicit

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()

frmoutput3prop1.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "0"

Text2.Text = "25"

Text3.Text = "50"

Text4.Text = "0"

Text9.Text = frmoutput3.Lstterminos.List(0)

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

frmoutput3.Lstterminos.List(0) = Text9.Text

If Text9.Text <> "term 1" Then

 frmoutput3prop1.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120

 If Text1.Text = "0" Then

 Y1 = 3120

 Else

 If Text1.Text = "1" Then

 Y1 = 120

 Else

 MsgBox "La Ordenada para este punto solo puede tener valor de cero o uno", vbExclamation,

"Atención"

 GoTo ignoredatos

 End If

```

End If
X2 = 120 + CInt(Val(Text2.Text)) * 50
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * 50
Y4 = 3120
frmoutput3.Line17(7).BorderColor = &HFF
frmoutput3.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120
    If Text1.Text = "0" Then
        Y1 = 3120
    Else
        If Text1.Text = "1" Then
            Y1 = 120
        Else
            MsgBox "La Ordenada para este punto solo puede tener valor de cero o uno", vbExclamation,
            "Atención"
            GoTo ignoredatos
        End If
    End If
End If
X2 = 120 + CInt(Val(Text2.Text)) * 50
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 3120
X4 = X3
Y4 = Y3
frmoutput3.Line17(7).BorderColor = &H0

```

```

    frmoutput3.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

With frmoutput3
    .Line17(1).X1 = X1
    .Line17(1).Y1 = Y1
    .Line17(1).X2 = X2
    .Line17(1).Y2 = Y2
    .Line17(1).Visible = True
    .Line17(2).X1 = X2
    .Line17(2).Y1 = Y2
    .Line17(2).X2 = X3
    .Line17(2).Y2 = Y3
    .Line17(2).Visible = True
    .Line17(3).X1 = X3
    .Line17(3).Y1 = Y3
    .Line17(3).X2 = X4
    .Line17(3).Y2 = Y4
    .Line17(3).Visible = True
    .Line17(4).X1 = X4
    .Line17(4).Y1 = Y4
    .Line17(4).X2 = 5120
    .Line17(4).Y2 = 3120

```

```

        .Line17(4).Visible = True
    End With
    frmoutput3prop1.Hide
    ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
    If Opttrapezoidal.Value = True Then
        Label4.Enabled = True
        Text4.Enabled = True
        Text8.Enabled = True
        Label8.Enabled = True
        Text7.Text = "1"
        Text1.Locked = False
    End If
End Sub

```

```

Private Sub opttriangular_Click()
    If opttriangular.Value = True Then
        Label4.Enabled = False
        Text4.Enabled = False
        Text8.Enabled = False
        Label8.Enabled = False
        Text1.Locked = True
        Text7.Text = "0"
    End If
End Sub

```

```

Private Sub Text9_Change()
    frmoutput3prop1.Caption = "Propiedades del término " + Text9.Text
End Sub

```

```

Option Explicit
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

```

```

Private Sub CancelButton_Click()
    frmoutput3prop2.Hide
End Sub

```

```
Private Sub Form_Load()
```

```
Text1.Text = "25"
```

```
Text2.Text = "50"
```

```
Text3.Text = "75"
```

```
Text4.Text = "0"
```

```
Text9.Text = frmoutput3.Lstterminos.List(1)
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
frmoutput3.Lstterminos.List(1) = Text9.Text
```

```
If Text9.Text <> "term 2" Then
```

```
    frmoutput3prop2.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * 50
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * 50
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * 50
```

```
    Y3 = 120
```

```
    X4 = 120 + CInt(Val(Text4.Text)) * 50
```

```
    Y4 = 3120
```

```
    frmoutput3.Line17(7).BorderColor = &HFF
```

```
    frmoutput3.Line17(8).BorderColor = &H0
```

```
    If X3 >= X4 Then
```

```
        MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,  
        "Atención"
```

```
        GoTo ignoredatos
```

```
    End If
```

```
End If
```

```
If opttriangular.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * 50
```

```

Y1 = 3120
X2 = 120 + CInt(Val(Text2.Text)) * 50
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 3120
X4 = X3
Y4 = Y3
frmoutput3.Line17(7).BorderColor = &H0
frmoutput3.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

With frmoutput3
    .Line17(5).X1 = X1
    .Line17(5).Y1 = Y1
    .Line17(5).X2 = X2
    .Line17(5).Y2 = Y2
    .Line17(5).Visible = True
    .Line17(6).X1 = X2
    .Line17(6).Y1 = Y2
    .Line17(6).X2 = X3
    .Line17(6).Y2 = Y3
    .Line17(6).Visible = True
    .Line17(7).X1 = X3

```

```

.Line17(7).Y1 = Y3
.Line17(7).X2 = X4
.Line17(7).Y2 = Y4
.Line17(7).Visible = True
.Line17(8).X1 = X4
.Line17(8).Y1 = Y4
.Line17(8).X2 = 5120
.Line17(8).Y2 = 3120
.Line17(8).Visible = True

```

```
End With
```

```
frmoutput3prop2.Hide
```

```
ignoredatos:
```

```
End Sub
```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text1.Locked = False
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmoutput3prop2.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

Option Explicit

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()

frmoutput3prop3.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "50"

Text2.Text = "75"

Text3.Text = "100"

Text4.Text = "1"

Text9.Text = frmoutput3.Lstterminos.List(2)

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

frmoutput3.Lstterminos.List(2) = Text9.Text

If Text9.Text <> "term 3" Then

 frmoutput3prop3.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * 50

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * 50

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * 50

 Y3 = 120

 X4 = 5120

 If Text4.Text = "0" Then

 Y4 = 3120

 frmoutput3.Line17(11).BorderColor = &HC000&

 frmoutput3.Line17(11).BorderStyle = 1

 Else

 If Text4.Text = "1" Then


```

    Y4 = 120
    frmoutput3.Line17(11).BorderColor = &H0&
    frmoutput3.Line17(11).BorderStyle = 3
Else
    MsgBox "La ordenada para el punto 3 solo puede tomar valor de cero o uno", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If
If X3 > X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If
If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frmoutput3.Line17(12).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
With frmoutput3
    .Line17(9).X1 = X1
    .Line17(9).Y1 = Y1
    .Line17(9).X2 = X2
    .Line17(9).Y2 = Y2

```

```

.Line17(9).Visible = True
.Line17(10).X1 = X2
.Line17(10).Y1 = Y2
.Line17(10).X2 = X3
.Line17(10).Y2 = Y3
.Line17(10).Visible = True
.Line17(11).X1 = X3
.Line17(11).Y1 = Y3
.Line17(11).X2 = X4
.Line17(11).Y2 = Y4
.Line17(11).Visible = True
.Line17(12).X1 = X4
.Line17(12).Y1 = Y4
.Line17(12).X2 = 5120
.Line17(12).Y2 = 3120
.Line17(12).Visible = True

```

End With

frmoutput3prop3.Hide

ignoredatos:

End Sub

Private Sub Opttrapezoidal_Click()

If Opttrapezoidal.Value = True Then

Text3.Locked = False

Text7.Text = "1"

Label4.Enabled = True

Text4.Enabled = True

Text8.Enabled = True

Label8.Enabled = True

Text4.Text = 1

End If

End Sub

Private Sub opttriangular_Click()

If opttriangular.Value = True Then

Text3.Locked = True

Text7.Text = "0"

```

Label4.Enabled = False
Text4.Enabled = False
Text8.Enabled = False
Label8.Enabled = False
End If
End Sub
Private Sub Text9_Change()
frmoutput3prop3.Caption = "Propiedades del término " + Text9.Text
End Sub

Option Explicit
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Private Sub CancelButton_Click()
frmoutput3prop4.Hide
End Sub

Private Sub Form_Load()
Text1.Text = "12.5"
Text2.Text = "25"
Text3.Text = "37.5"
Text4.Text = "0"
Text9.Text = frmoutput3.Lstterminos.List(3)
opttriangular.Value = True
Opttrapezoidal.Value = False
End Sub

Private Sub OKButton_Click()
frmoutput3.Lstterminos.List(3) = Text9.Text
If Text9.Text <> "term 4" Then
    frmoutput3prop4.Caption = "Propiedades de " + Text9.Text
End If
If Opttrapezoidal.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120

```

```

X3 = 120 + CInt(Val(Text3.Text)) * 50
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * 50
Y4 = 3120
frmoutput3.Line17(7).BorderColor = &HFF
frmoutput3.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frmoutput3.Line17(7).BorderColor = &H0
    frmoutput3.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then

```

```

        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If
With frmoutput3
    .Line17(13).X1 = X1
    .Line17(13).Y1 = Y1
    .Line17(13).X2 = X2
    .Line17(13).Y2 = Y2
    .Line17(13).Visible = True
    .Line17(14).X1 = X2
    .Line17(14).Y1 = Y2
    .Line17(14).X2 = X3
    .Line17(14).Y2 = Y3
    .Line17(14).Visible = True
    .Line17(15).X1 = X3
    .Line17(15).Y1 = Y3
    .Line17(15).X2 = X4
    .Line17(15).Y2 = Y4
    .Line17(15).Visible = True
    .Line17(16).X1 = X4
    .Line17(16).Y1 = Y4
    .Line17(16).X2 = 5120
    .Line17(16).Y2 = 3120
    .Line17(16).Visible = True
End With
frmoutput3prop4.Hide
ignoredatos:
End Sub

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"

```

```

    Text1.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```

Private Sub Text9_Change()
frmoutput3prop4.Caption = "Propiedades del término " + Text9.Text
End Sub

```

Option Explicit

```

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

```

```

Private Sub CancelButton_Click()
frmoutput3prop5.Hide
End Sub

```

```

Private Sub Form_Load()
Text1.Text = "62.5"
Text2.Text = "75"
Text3.Text = "87.5"
Text4.Text = "0"
Text9.Text = frmoutput3.Lstterminos.List(4)
opttriangular.Value = True
Opttrapezoidal.Value = False
End Sub

```

```

Private Sub OKButton_Click()
frmoutput3.Lstterminos.List(4) = Text9.Text
If Text9.Text <> "term 5" Then

```

```

    frmoutput3prop5.Caption = "Propiedades de " + Text9.Text
End If
If Opttrapezoidal.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 120
    X4 = 120 + CInt(Val(Text4.Text)) * 50
    Y4 = 3120
    frmoutput3.Line17(7).BorderColor = &HFF
    frmoutput3.Line17(8).BorderColor = &H0
    If X3 >= X4 Then
        MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * 50
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * 50
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * 50
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frmoutput3.Line17(7).BorderColor = &H0
    frmoutput3.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"

```

```

    GoTo ignoredatos
End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de 100", vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If
With frmoutput3
    .Line17(17).X1 = X1
    .Line17(17).Y1 = Y1
    .Line17(17).X2 = X2
    .Line17(17).Y2 = Y2
    .Line17(17).Visible = True
    .Line17(18).X1 = X2
    .Line17(18).Y1 = Y2
    .Line17(18).X2 = X3
    .Line17(18).Y2 = Y3
    .Line17(18).Visible = True
    .Line17(19).X1 = X3
    .Line17(19).Y1 = Y3
    .Line17(19).X2 = X4
    .Line17(19).Y2 = Y4
    .Line17(19).Visible = True
    .Line17(20).X1 = X4
    .Line17(20).Y1 = Y4
    .Line17(20).X2 = 5120
    .Line17(20).Y2 = 3120
    .Line17(20).Visible = True
End With
frmoutput3prop5.Hide
ignoredatos:
End Sub

```



```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text1.Locked = False
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmoutput3prop5.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Dim w As Variant
```

```
Dim lazo, i, temporal, row, column As Integer
```

```
Dim fila, columna As Integer
```

```
Private Sub cmdaceptar_Click()
```

```
frmruleblock1.Hide
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
With MSFlexGrid1
```

```
    .Rows = matriz_indices(1, 2)
```

```
For lazo = 0 To .Rows - 1
```

```
    .RowHeight(lazo) = 350
```

```

Next lazo
If matriz_indices(2, 2) <> 0 Then
    .Cols = matriz_indices(2, 2)
Else
    .Cols = 1
End If
For lazo = 0 To .Cols - 1
    .ColWidth(lazo) = 800
Next
.Height = .RowHeight(0) * .Rows
.Width = .ColWidth(0) * .Cols
.Left = frmruleblock1.Width - (Frame1.Left + Frame1.Width + (Text2(0).Width + .Width) / 2)
.Top = (Frame2.Top - .Height) / 2
If .Cols > 1 Then
    For lazo = 0 To .Cols - 1
        Text3(lazo).Top = .Top + .Height + Text3(lazo).Height
        Text3(lazo).Left = .Left + 20 + Text3(lazo).Width * lazo
        Text3(lazo).Text = Combo1(2).List(lazo)
        Text3(lazo).Visible = True
    Next lazo
    Text5.Left = .Left + (.Width - Text5.Width) / 2 + 20
    Text5.Top = Text3(0).Top + Text3(0).Height
    Text5.Text = Text1(2).Text
    Text5.Visible = True
End If
End With
For lazo = 0 To MSFlexGrid1.Rows - 1
    Text2(lazo).Top = MSFlexGrid1.Top + 40 + Text2(lazo).Height * lazo
    Text2(lazo).Left = MSFlexGrid1.Left - Text2(lazo).Width
    Text2(lazo).Text = Combo1(1).List(lazo)
    Text2(lazo).Visible = True
Next lazo
Text4.Left = MSFlexGrid1.Left - Text4.Width - 50
Text4.Top = Text2(0).Top - Text4.Height
Text4.Text = Text1(1).Text
Text4.Visible = True

```

End Sub

Private Sub Form_Load()

fila = 0

columna = 0

Set tbrdefin1.ImageList = ImageList1

With tbrdefin1.Buttons

.Add 1, "Debug", , tbrDefault, "Debug"

.Add 1, "Rules", , tbrDefault, "Rules"

.Item(2).ToolTipText = "Depurar"

.Item(1).ToolTipText = "Generar reglas"

End With

End Sub

Private Sub MSFlexGrid1_Click()

fila = MSFlexGrid1.row

columna = MSFlexGrid1.Col

Combo1(1).ListIndex = fila

If input_index >= 2 Then Combo1(2).ListIndex = columna

MSFlexGrid1.TextMatrix(fila, columna) = Combo6(1).List(Combo6(1).ListIndex)

End Sub

Private Sub MSFlexGrid1_DblClick()

With MSFlexGrid1

.TextMatrix(.row, .Col) = ""

End With

End Sub

Private Sub tbrdefin1_ButtonClick(ByVal Button As MSComctlLib.Button)

Select Case Button.Key

Case "Rules"

MSFlexGrid2.Clear

'Text7.Text = FuzzyAnd(entrada1(1), entrada2(1))

i = 0

If input_index = 1 Then

For row = 0 To MSFlexGrid1.Rows - 1

If MSFlexGrid1.TextMatrix(row, 0) = "" Then

```

        GoTo otra_regla
    Else
        For lazo = 0 To Combo6(1).ListCount - 1
            If MSFlexGrid1.TextMatrix(row, 0) = Combo6(1).List(lazo) Then
                matriz_puntero(i, 0) = row
                MSFlexGrid2.TextMatrix(i, 0) = row
                matriz_puntero(i, 5) = lazo
                MSFlexGrid2.TextMatrix(i, 2) = lazo
            End If
        Next lazo
        i = i + 1
    otra_regla:
        End If
    Next row
End If

If input_index = 2 Then
'Text7.Text = FuzzyAnd(entrada3(1), FuzzyAnd(entrada1(1), entrada2(1)))
For column = 0 To MSFlexGrid1.Rows - 1
    For row = 0 To MSFlexGrid1.Cols - 1
        If MSFlexGrid1.TextMatrix(row, column) = "" Then
            GoTo siguiente
        Else
            For lazo = 0 To Combo6(1).ListCount - 1
                If MSFlexGrid1.TextMatrix(row, column) = Combo6(1).List(lazo) Then
                    matriz_puntero(i, 0) = row
                    'MSFlexGrid2.TextMatrix(i, 0) = row
                    matriz_puntero(i, 1) = column
                    'MSFlexGrid2.TextMatrix(i, 1) = column
                    matriz_puntero(i, 5) = lazo
                    'MSFlexGrid2.TextMatrix(i, 2) = lazo
                End If
            Next lazo
            i = i + 1
        siguiente:
            End If
    Next row

```

Next column

End If

Case "Debug"

MSFlexGrid2.Clear

monitor1.Show

End Select

End Sub

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino1prop.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "1"

Text2.Text = "0"

Text3.Text = "25"

Text4.Text = "50"

Text9.Text = frminput1.Lstterminos.List(0)

opttriangular.Value = False

Opttrapezoidal.Value = True

Text5.Text = Universo_entrada(1, 1)

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(1, 2) - Universo_entrada(1, 1)

paso = Universo / 255

frminput1.Lstterminos.List(0) = Text9.Text

If Text9.Text <> "term 1" Then

frmtermino1prop.Caption = "Propiedades de " + Text9.Text

End If

```

If Opttrapezoidal.Value = True Then
    X1 = 120
    If CInt(Val(Text1.Text)) = 0 Then
        Y1 = 3120
    Else
        If CInt(Val(Text1.Text)) = 1 Then
            Y1 = 120
        Else
            MsgBox "La ordenada para el punto 1 solo puede tomar valor de cero o uno", vbExclamation,
            "Atención"
            GoTo ignoredatos
        End If
    End If
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 120
    X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
    Y4 = 3120
    If X3 >= X4 Then
        MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If

End If

If opttriangular.Value = True Then
    X1 = 120
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
End If

```

```

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
With frminput1
    .Line17(1).X1 = X1
    .Line17(1).Y1 = Y1
    .Line17(1).X2 = X2
    .Line17(1).Y2 = Y2
    .Line17(1).Visible = True
    .Line17(2).X1 = X2
    .Line17(2).Y1 = Y2
    .Line17(2).X2 = X3
    .Line17(2).Y2 = Y3
    .Line17(2).Visible = True
    .Line17(3).X1 = X3
    .Line17(3).Y1 = Y3
    .Line17(3).X2 = X4
    .Line17(3).Y2 = Y4
    .Line17(3).Visible = True
    .Line17(4).X1 = X4
    .Line17(4).Y1 = Y4
    .Line17(4).X2 = 5120
    .Line17(4).Y2 = 3120
    .Line17(4).Visible = True
End With
If X2 > 120 Then
    frminput1.Line17(1).BorderColor = &HC00000
Else
    frminput1.Line17(1).BorderColor = &H0
End If

'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255

```

```

Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z4 - Z3))
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z4 - Z3))
    End If
    Punto_pendiente(1, 2) = Z4
End If
If opttriangular.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z3 - Z2))
    Else
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(1, 2) = Z3
End If
Punto_pendiente(1, 1) = Z0
Punto_pendiente(1, 3) = M1
Punto_pendiente(1, 4) = M2

frmtermino1prop.Hide
frminput1.Show
ignoredatos:
End Sub

```



```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text1.Text = "1"
```

```
    Text1.Locked = False
```

```
    Text5.Text = Universo_entrada(1, 1)
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text4.Text = Text3.Text
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
    Text1.Text = "0"
```

```
    Text1.Locked = True
```

```
    Text5.Text = Universo_entrada(1, 1)
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmtermino1prop.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Option Explicit
```

```
Dim temporal, temporal1 As Variant
```

```
Dim Universo, paso As Variant
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
```

```
frmtermino1prop2.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "1"
```

```
Text2.Text = "0"
```

```
Text3.Text = "25"
```

```
Text4.Text = "50"
```

```
Text9.Text = frminput2.Lstterminos.List(0)
```

```
opttriangular.Value = False
```

```
Opttrapezoidal.Value = True
```

```
Text5.Text = Universo_entrada(2, 1)
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(2, 2) - Universo_entrada(2, 1)
```

```
paso = Universo / 255
```

```
frminput2.Lstterminos.List(0) = Text9.Text
```

```
If frminput2.Lstterminos.List(0) <> "term 1" Then
```

```
    frmtermino1prop2.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120
```

```
    If CInt(Val(Text1.Text)) = 0 Then
```

```
        Y1 = 3120
```

```
    Else
```

```
        If CInt(Val(Text1.Text)) = 1 Then
```

```
            Y1 = 120
```

```
        Else
```

```
            MsgBox "La ordenada para el punto 1 solo puede tomar valor de cero o uno", vbExclamation,  
"Atención"
```

```
            GoTo ignoredatos
```

```
        End If
```

```
    End If
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```

Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

End If

If opttriangular.Value = True Then
    X1 = 120
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

With frminput2
    .Line17(1).X1 = X1
    .Line17(1).Y1 = Y1
    .Line17(1).X2 = X2
    .Line17(1).Y2 = Y2
    .Line17(1).Visible = True
    .Line17(2).X1 = X2
    .Line17(2).Y1 = Y2
    .Line17(2).X2 = X3

```

```

.Line17(2).Y2 = Y3
.Line17(2).Visible = True
.Line17(3).X1 = X3
.Line17(3).Y1 = Y3
.Line17(3).X2 = X4
.Line17(3).Y2 = Y4
.Line17(3).Visible = True
.Line17(4).X1 = X4
.Line17(4).Y1 = Y4
.Line17(4).X2 = 5120
.Line17(4).Y2 = 3120
.Line17(4).Visible = True
End With

If X2 > 120 Then
    frminput2.Line17(1).BorderColor = &HC00000
Else
    frminput2.Line17(1).BorderColor = &H0
End If

'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z4 - Z3))
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z4 - Z3))
    End If
End If

```

```

End If
Punto_pendiente(2, 2) = Z4
End If
If opttriangular.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z3 - Z2))
    Else
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(2, 2) = Z3
End If

Punto_pendiente(2, 1) = Z0
Punto_pendiente(2, 3) = M1
Punto_pendiente(2, 4) = M2

frmtermino1prop2.Hide
frminput2.Show
ignoredatos:
End Sub

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Text = "1"
    Text1.Locked = False
    Text5.Text = Universo_entrada(2, 1)
End If
End Sub

```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
    Text1.Text = "0"
```

```
    Text1.Locked = True
```

```
    Text5.Text = Universo_entrada(2, 1)
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmtermino1prop2.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Option Explicit
```

```
Dim temporal, temporal1 As Variant
```

```
Dim Universo, paso As Variant
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
```

```
frmtermino1prop3.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "1"
```

```
Text2.Text = "0"
```

```
Text3.Text = "25"
```

```
Text4.Text = "50"
```

```
Text9.Text = frminput3.Lstterminos.List(0)
```

```
opttriangular.Value = False
```

```
Opttrapezoidal.Value = True
```

```
Text5.Text = Universo_entrada(3, 1)
```

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(3, 2) - Universo_entrada(3, 1)

paso = Universo / 255

frminput3.Lstterminos.List(0) = Text9.Text

If frminput3.Lstterminos.List(0) <> "term 1" Then

frmtermino1prop3.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

X1 = 120

If CInt(Val(Text1.Text)) = 0 Then

Y1 = 3120

Else

If CInt(Val(Text1.Text)) = 1 Then

Y1 = 120

Else

MsgBox "La ordenada para el punto 1 solo puede tomar valor de cero o uno", vbExclamation,
"Atención"

GoTo ignoredatos

End If

End If

X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

Y2 = 120

X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

Y3 = 120

X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)

Y4 = 3120

If X3 >= X4 Then

MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
"Atención"

GoTo ignoredatos

End If

End If

If opttriangular.Value = True Then

X1 = 120

```

Y1 = 3120
X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
Y3 = 3120
X4 = X3
Y4 = Y3
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

With frminput3
    .Line17(1).X1 = X1
    .Line17(1).Y1 = Y1
    .Line17(1).X2 = X2
    .Line17(1).Y2 = Y2
    .Line17(1).Visible = True
    .Line17(2).X1 = X2
    .Line17(2).Y1 = Y2
    .Line17(2).X2 = X3
    .Line17(2).Y2 = Y3
    .Line17(2).Visible = True
    .Line17(3).X1 = X3
    .Line17(3).Y1 = Y3
    .Line17(3).X2 = X4
    .Line17(3).Y2 = Y4
    .Line17(3).Visible = True
    .Line17(4).X1 = X4
    .Line17(4).Y1 = Y4
    .Line17(4).X2 = 5120
    .Line17(4).Y2 = 3120
    .Line17(4).Visible = True
End With

If X2 > 120 Then

```



```

    frminput3.Line17(1).BorderColor = &HC00000
Else
    frminput3.Line17(1).BorderColor = &H0
End If

'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z4 - Z3))
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z4 - Z3))
    End If
    Punto_pendiente(3, 2) = Z4
End If
If opttriangular.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z3 - Z2))
    Else
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(3, 2) = Z3
End If
Punto_pendiente(3, 1) = Z0

```

Punto_pendiente(3, 3) = M1

Punto_pendiente(3, 4) = M2

frmtermino1prop3.Hide

frminput3.Show

ignoredatos:

End Sub

Private Sub Opttrapezoidal_Click()

If Opttrapezoidal.Value = True Then

Label4.Enabled = True

Text4.Enabled = True

Text8.Enabled = True

Label8.Enabled = True

Text7.Text = "1"

Text1.Text = "1"

Text1.Locked = False

Text5.Text = Universo_entrada(3, 1)

End If

End Sub

Private Sub opttriangular_Click()

If opttriangular.Value = True Then

Label4.Enabled = False

Text4.Enabled = False

Text8.Enabled = False

Label8.Enabled = False

Text7.Text = "0"

Text1.Text = "0"

Text1.Locked = True

Text5.Text = Universo_entrada(3, 1)

End If

End Sub

Private Sub Text9_Change()

frmtermino1prop3.Caption = "Propiedades del término " + Text9.Text

End Sub

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino1prop4.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "1"

Text2.Text = "0"

Text3.Text = "25"

Text4.Text = "50"

Text9.Text = frminput4.Lstterminos.List(0)

opttriangular.Value = False

Opttrapezoidal.Value = True

Text5.Text = Universo_entrada(4, 1)

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(4, 2) - Universo_entrada(4, 1)

paso = Universo / 255

frminput4.Lstterminos.List(0) = Text9.Text

If frminput4.Lstterminos.List(0) <> "term 1" Then

 frmtermino1prop4.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120

 If CInt(Val(Text1.Text)) = 0 Then

 Y1 = 3120

 Else

 If CInt(Val(Text1.Text)) = 1 Then

 Y1 = 120

 Else

```

        MsgBox "La ordenada para el punto 1 solo puede tomar valor de cero o uno", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If
X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

End If
If opttriangular.Value = True Then
    X1 = 120
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
With frminput4
    .Line17(1).X1 = X1
    .Line17(1).Y1 = Y1

```

```

.Line17(1).X2 = X2
.Line17(1).Y2 = Y2
.Line17(1).Visible = True
.Line17(2).X1 = X2
.Line17(2).Y1 = Y2
.Line17(2).X2 = X3
.Line17(2).Y2 = Y3
.Line17(2).Visible = True
.Line17(3).X1 = X3
.Line17(3).Y1 = Y3
.Line17(3).X2 = X4
.Line17(3).Y2 = Y4
.Line17(3).Visible = True
.Line17(4).X1 = X4
.Line17(4).Y1 = Y4
.Line17(4).X2 = 5120
.Line17(4).Y2 = 3120
.Line17(4).Visible = True
End With
If X2 > 120 Then
    frminput4.Line17(1).BorderColor = &HC00000
Else
    frminput4.Line17(1).BorderColor = &H0
End If
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z4 - Z3))
    Else
        'temporal1 = temporal / paso

```

```

    'If (temporal1 - Fix(temporal1)) <= 0.5 Then
    '    temporal1 = Fix(temporal1)
    'End If
    M1 = Fix(&HFF / (Z2 - Z0))
    M2 = Fix(&HFF / (Z4 - Z3))
End If
Punto_pendiente(4, 2) = Z4
End If
If opttriangular.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z3 - Z2))
    Else
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(4, 2) = Z3
End If
Punto_pendiente(4, 1) = Z0
Punto_pendiente(4, 3) = M1
Punto_pendiente(4, 4) = M2

```

```

frmtermino1prop4.Hide
frminput4.Show
ignoredatos:
End Sub

```

```
Private Sub Opttrapezoidal_Click()
```

```

If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Text = "1"

```

```

Text1.Locked = False
Text5.Text = Universo_entrada(4, 1)
End If
End Sub

```

```

Private Sub opttriangular_Click()

```

```

If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
    Text1.Text = "0"
    Text1.Locked = True
    Text5.Text = Universo_entrada(4, 1)
End If
End Sub

```

```

Private Sub Text9_Change()

```

```

frmtermino1prop4.Caption = "Propiedades del término " + Text9.Text
End Sub

```

```

Option Explicit

```

```

Dim temporal, temporal1 As Variant
Dim Universo, paso As Variant
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

```

```

Private Sub CancelButton_Click()

```

```

frmtermino1prop5.Hide
End Sub

```

```

Private Sub Form_Load()

```

```

Text1.Text = "1"
Text2.Text = "0"
Text3.Text = "25"
Text4.Text = "50"
Text9.Text = frminput5.Lstterminos.List(0)
opttriangular.Value = False

```

```
Opttrapezoidal.Value = True
```

```
Text5.Text = Universo_entrada(5, 1)
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(5, 2) - Universo_entrada(5, 1)
```

```
paso = Universo / 255
```

```
frminput5.Lstterminos.List(0) = Text9.Text
```

```
If frminput5.Lstterminos.List(0) <> "term 1" Then
```

```
    frmtermino1prop5.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120
```

```
    If CInt(Val(Text1.Text)) = 0 Then
```

```
        Y1 = 3120
```

```
    Else
```

```
        If CInt(Val(Text1.Text)) = 1 Then
```

```
            Y1 = 120
```

```
        Else
```

```
            MsgBox "La ordenada para el punto 1 solo puede tomar valor de cero o uno", vbExclamation,
```

```
"Atención"
```

```
            GoTo ignoredatos
```

```
        End If
```

```
    End If
```

```
X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```
Y2 = 120
```

```
X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
```

```
Y3 = 120
```

```
X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
```

```
Y4 = 3120
```

```
If X3 >= X4 Then
```

```
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
```

```
"Atención"
```

```
    GoTo ignoredatos
```

```
End If
```

```
End If
```


If opttriangular.Value = True Then

 X1 = 120

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 3120

 X4 = X3

 Y4 = Y3

End If

If X2 >= X3 Then

 MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
 "Atención"

 GoTo ignoredatos

End If

With frminput5

 .Line17(1).X1 = X1

 .Line17(1).Y1 = Y1

 .Line17(1).X2 = X2

 .Line17(1).Y2 = Y2

 .Line17(1).Visible = True

 .Line17(2).X1 = X2

 .Line17(2).Y1 = Y2

 .Line17(2).X2 = X3

 .Line17(2).Y2 = Y3

 .Line17(2).Visible = True

 .Line17(3).X1 = X3

 .Line17(3).Y1 = Y3

 .Line17(3).X2 = X4

 .Line17(3).Y2 = Y4

 .Line17(3).Visible = True

 .Line17(4).X1 = X4

 .Line17(4).Y1 = Y4

 .Line17(4).X2 = 5120

 .Line17(4).Y2 = 3120

 .Line17(4).Visible = True

```

End With
If X2 > 120 Then
    frminput5.Line17(1).BorderColor = &HC00000
Else
    frminput5.Line17(1).BorderColor = &H0
End If
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = &HFF / (Z4 - Z3)
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z4 - Z3))
    End If
    Punto_pendiente(5, 2) = Z4
End If
If opttriangular.Value = True Then
    temporal = Z2 - Z0
    If temporal = 0 Then
        M1 = &H0
        M2 = Fix(&HFF / (Z3 - Z2))
    Else
        M1 = Fix(&HFF / (Z2 - Z0))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(5, 2) = Z3

```

End If

Punto_pendiente(5, 1) = Z0

Punto_pendiente(5, 3) = M1

Punto_pendiente(5, 4) = M2

frmtermino1prop5.Hide

frminput5.Show

ignoredatos:

End Sub

Private Sub Opttrapezoidal_Click()

If Opttrapezoidal.Value = True Then

Label4.Enabled = True

Text4.Enabled = True

Text8.Enabled = True

Label8.Enabled = True

Text7.Text = "1"

Text1.Text = "1"

Text1.Locked = False

Text5.Text = Universo_entrada(5, 1)

End If

End Sub

Private Sub opttriangular_Click()

If opttriangular.Value = True Then

Label4.Enabled = False

Text4.Enabled = False

Text8.Enabled = False

Label8.Enabled = False

Text7.Text = "0"

Text1.Text = "0"

Text1.Locked = True

Text5.Text = Universo_entrada(5, 1)

End If

End Sub

Private Sub Text9_Change()

```
frmtermino1prop5.Caption = "Propiedades del término " + Text9.Text
End Sub
```

```
Option Explicit
Dim temporal, temporal1 As Variant
Dim Universo, paso As Variant
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
frmtermino2prop.Hide
End Sub
```

```
Private Sub Form_Load()
Text1.Text = "25"
Text2.Text = "50"
Text3.Text = "75"
Text4.Text = "0"
Text9.Text = frminput1.Lstterminos.List(1)
opttriangular.Value = True
Opttrapezoidal.Value = False
End Sub
```

```
Private Sub OKButton_Click()
Universo = Universo_entrada(1, 2) - Universo_entrada(1, 1)
paso = Universo / 255
frminput1.Lstterminos.List(1) = Text9.Text
If Text9.Text <> "term 2" Then
    frmtermino2prop.Caption = "Propiedades de " + Text9.Text
End If
If Opttrapezoidal.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 120
```

```

X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
frminput1.Line17(7).BorderColor = &HFF
frminput1.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput1.Line17(7).BorderColor = &H0
    frminput1.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(1, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then

```

```
MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(1, 2)),  
vbExclamation, "Atención"
```

```
GoTo ignoredatos
```

```
End If
```

```
End If
```

```
With frminput1
```

```
.Line17(5).X1 = X1
```

```
.Line17(5).Y1 = Y1
```

```
.Line17(5).X2 = X2
```

```
.Line17(5).Y2 = Y2
```

```
.Line17(5).Visible = True
```

```
.Line17(6).X1 = X2
```

```
.Line17(6).Y1 = Y2
```

```
.Line17(6).X2 = X3
```

```
.Line17(6).Y2 = Y3
```

```
.Line17(6).Visible = True
```

```
.Line17(7).X1 = X3
```

```
.Line17(7).Y1 = Y3
```

```
.Line17(7).X2 = X4
```

```
.Line17(7).Y2 = Y4
```

```
.Line17(7).Visible = True
```

```
.Line17(8).X1 = X4
```

```
.Line17(8).Y1 = Y4
```

```
.Line17(8).X2 = 5120
```

```
.Line17(8).Y2 = 3120
```

```
.Line17(8).Visible = True
```

```
End With
```

```
'calculo de pendientes
```

```
Z0 = (Val(Text5.Text) / Universo) * 255
```

```
Z1 = (Val(Text1.Text) / Universo) * 255
```

```
Z2 = (Val(Text2.Text) / Universo) * 255
```

```
Z3 = (Val(Text3.Text) / Universo) * 255
```

```
Z4 = (Val(Text4.Text) / Universo) * 255
```

```
If Opttrapezoidal.Value = True Then
```

```
M1 = Fix(&HFF / (Z2 - Z1))
```

```

    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(6, 2) = Z4
End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z3 - Z2))
    Punto_pendiente(6, 2) = Z3
End If

```

```

Punto_pendiente(6, 1) = Z1
Punto_pendiente(6, 3) = M1
Punto_pendiente(6, 4) = M2

```

```

frmtermino2prop.Hide
frminput1.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If

```

End Sub

Private Sub Text9_Change()

frmtermino2prop.Caption = "Propiedades del término " + Text9.Text

End Sub

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino2prop2.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "25"

Text2.Text = "50"

Text3.Text = "75"

Text4.Text = "0"

Text9.Text = frminput2.Lstterminos.List(1)

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(2, 2) - Universo_entrada(2, 1)

paso = Universo / 255

frminput2.Lstterminos.List(1) = Text9.Text

If frminput2.Lstterminos.List(1) <> "term 2" Then

 frmtermino2prop2.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120


```

X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
frminput2.Line17(7).BorderColor = &HFF
frminput2.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput2.Line17(7).BorderColor = &H0
    frminput2.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(2, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else

```

```

If Opttrapezoidal.Value = True And X4 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(2, 2)),
vbExclamation, "Atención"
    GoTo ignoredatos
End If
End If
With frminput2
    .Line17(5).X1 = X1
    .Line17(5).Y1 = Y1
    .Line17(5).X2 = X2
    .Line17(5).Y2 = Y2
    .Line17(5).Visible = True
    .Line17(6).X1 = X2
    .Line17(6).Y1 = Y2
    .Line17(6).X2 = X3
    .Line17(6).Y2 = Y3
    .Line17(6).Visible = True
    .Line17(7).X1 = X3
    .Line17(7).Y1 = Y3
    .Line17(7).X2 = X4
    .Line17(7).Y2 = Y4
    .Line17(7).Visible = True
    .Line17(8).X1 = X4
    .Line17(8).Y1 = Y4
    .Line17(8).X2 = 5120
    .Line17(8).Y2 = 3120
    .Line17(8).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))

```

```

    Punto_pendiente(7, 2) = Z4
End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z3 - Z2))
    Punto_pendiente(7, 2) = Z3
End If

```

```

Punto_pendiente(7, 1) = Z1
Punto_pendiente(7, 3) = M1
Punto_pendiente(7, 4) = M2

```

```

frmtermino2prop2.Hide
frminput2.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```
Private Sub Text9_Change()
```

```
frmtermino2prop2.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Option Explicit
```

```
Dim temporal, temporal1 As Variant
```

```
Dim Universo, paso As Variant
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
```

```
frmtermino2prop3.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "25"
```

```
Text2.Text = "50"
```

```
Text3.Text = "75"
```

```
Text4.Text = "0"
```

```
Text9.Text = frminput3.Lstterminos.List(1)
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(3, 2) - Universo_entrada(3, 1)
```

```
paso = Universo / 255
```

```
frminput3.Lstterminos.List(1) = Text9.Text
```

```
If frminput3.Lstterminos.List(1) <> "term 2" Then
```

```
    frmtermino2prop3.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
```

```

Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
frminput3.Line17(7).BorderColor = &HFF
frminput3.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput3.Line17(7).BorderColor = &H0
    frminput3.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(3, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then

```

```

    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(3, 2)),
vbExclamation, "Atención"
    GoTo ignoredatos
End If
End If
With frminput3
    .Line17(5).X1 = X1
    .Line17(5).Y1 = Y1
    .Line17(5).X2 = X2
    .Line17(5).Y2 = Y2
    .Line17(5).Visible = True
    .Line17(6).X1 = X2
    .Line17(6).Y1 = Y2
    .Line17(6).X2 = X3
    .Line17(6).Y2 = Y3
    .Line17(6).Visible = True
    .Line17(7).X1 = X3
    .Line17(7).Y1 = Y3
    .Line17(7).X2 = X4
    .Line17(7).Y2 = Y4
    .Line17(7).Visible = True
    .Line17(8).X1 = X4
    .Line17(8).Y1 = Y4
    .Line17(8).X2 = 5120
    .Line17(8).Y2 = 3120
    .Line17(8).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(8, 2) = Z4

```

```

End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z3 - Z2))
    Punto_pendiente(8, 2) = Z3
End If

```

```

Punto_pendiente(8, 1) = Z1
Punto_pendiente(8, 3) = M1
Punto_pendiente(8, 4) = M2

```

```

frmtermino2prop3.Hide
frminput3.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub
Private Sub Text9_Change()

```

```
frmtermino2prop3.Caption = "Propiedades del término " + Text9.Text
End Sub
```

```
Option Explicit
Dim temporal, temporal1 As Variant
Dim Universo, paso As Variant
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
Private Sub CancelButton_Click()
frmtermino2prop4.Hide
End Sub
```

```
Private Sub Form_Load()
Text1.Text = "25"
Text2.Text = "50"
Text3.Text = "75"
Text4.Text = "0"
Text9.Text = frminput4.Lstterminos.List(1)
opttriangular.Value = True
Opttrapezoidal.Value = False
End Sub
```

```
Private Sub OKButton_Click()
Universo = Universo_entrada(4, 2) - Universo_entrada(4, 1)
paso = Universo / 255
frminput4.Lstterminos.List(1) = Text9.Text
If frminput4.Lstterminos.List(1) <> "term 2" Then
    frmtermino2prop4.Caption = "Propiedades de " + Text9.Text
End If
If Opttrapezoidal.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 120
    X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
```



```

Y4 = 3120
frminput4.Line17(7).BorderColor = &HFF
frminput4.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput4.Line17(7).BorderColor = &H0
    frminput4.Line17(8).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
        vbExclamation, "Atención"
    
```

```

        GoTo ignoredatos
    End If
End If
With frminput4
    .Line17(5).X1 = X1
    .Line17(5).Y1 = Y1
    .Line17(5).X2 = X2
    .Line17(5).Y2 = Y2
    .Line17(5).Visible = True
    .Line17(6).X1 = X2
    .Line17(6).Y1 = Y2
    .Line17(6).X2 = X3
    .Line17(6).Y2 = Y3
    .Line17(6).Visible = True
    .Line17(7).X1 = X3
    .Line17(7).Y1 = Y3
    .Line17(7).X2 = X4
    .Line17(7).Y2 = Y4
    .Line17(7).Visible = True
    .Line17(8).X1 = X4
    .Line17(8).Y1 = Y4
    .Line17(8).X2 = 5120
    .Line17(8).Y2 = 3120
    .Line17(8).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(9, 2) = Z4
End If
If opttriangular.Value = True Then

```

```

M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(9, 2) = Z3
End If

```

```

Punto_pendiente(9, 1) = Z1
Punto_pendiente(9, 3) = M1
Punto_pendiente(9, 4) = M2

```

```

frmtermino2prop4.Hide
frminput4.Show
ignoredatos:
End Sub

```

```
Private Sub Opttrapezoidal_Click()
```

```

If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Locked = False
End If
End Sub

```

```
Private Sub opttriangular_Click()
```

```

If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```
Private Sub Text9_Change()
```

```

frmtermino2prop4.Caption = "Propiedades del término " + Text9.Text
End Sub

```

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

[Private Sub CancelButton_Click\(\)](#)

frmtermino2prop5.Hide

End Sub

Private Sub Form_Load()

Text1.Text = "25"

Text2.Text = "50"

Text3.Text = "75"

Text4.Text = "0"

Text9.Text = frminput5.Lstterminos.List(1)

opttriangular.Value = True

Opttrapezoidal.Value = False

[End Sub](#)

[Private Sub OKButton_Click\(\)](#)

Universo = Universo_entrada(5, 2) - Universo_entrada(5, 1)

paso = Universo / 255

frminput5.Lstterminos.List(1) = Text9.Text

If frminput5.Lstterminos.List(1) <> "term 2" Then

 frmtermino2prop5.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120

 X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)

 Y4 = 3120

```

frminput5.Line17(7).BorderColor = &HFF
frminput5.Line17(8).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

```

```

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput5.Line17(7).BorderColor = &H0
    frminput5.Line17(8).BorderColor = &H0
End If

```

```

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

```

```

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(5, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(5, 2)),
        vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

```

```

    End If
End If
With frminput5
    .Line17(5).X1 = X1
    .Line17(5).Y1 = Y1
    .Line17(5).X2 = X2
    .Line17(5).Y2 = Y2
    .Line17(5).Visible = True
    .Line17(6).X1 = X2
    .Line17(6).Y1 = Y2
    .Line17(6).X2 = X3
    .Line17(6).Y2 = Y3
    .Line17(6).Visible = True
    .Line17(7).X1 = X3
    .Line17(7).Y1 = Y3
    .Line17(7).X2 = X4
    .Line17(7).Y2 = Y4
    .Line17(7).Visible = True
    .Line17(8).X1 = X4
    .Line17(8).Y1 = Y4
    .Line17(8).X2 = 5120
    .Line17(8).Y2 = 3120
    .Line17(8).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(10, 2) = Z4
End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))

```

```

M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(10, 2) = Z3
End If

```

```

Punto_pendiente(10, 1) = Z1
Punto_pendiente(10, 3) = M1
Punto_pendiente(10, 4) = M2

```

```

frmtermino2prop5.Hide
frminput5.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text1.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```

Private Sub Text9_Change()
frmtermino2prop5.Caption = "Propiedades del término " + Text9.Text
End Sub

```

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z5, M1, M2 As Byte

Private Sub CancelButton_Click()

If Text3.Text <> Text8.Text And Text4.Text = "1" Then

 MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
vbExclamation, "Atención"

 GoTo ignorecancelar

End If

frmtermino3prop.Hide

ignorecancelar:

End Sub

Private Sub Form_Load()

Text1.Text = "50"

Text2.Text = "75"

Text3.Text = "100"

Text4.Text = "1"

Text8.Text = Universo_entrada(1, 2)

Text9.Text = frminput1.Lstterminos.List(2)

opttriangular.Value = False

Opttrapezoidal.Value = True

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(1, 2) - Universo_entrada(1, 1)

paso = Universo / 255

frminput1.Lstterminos.List(2) = Text9.Text

If Text9.Text <> "term 3" Then

 frmtermino3prop.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120


```
X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```
Y2 = 120
```

```
X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
```

```
Y3 = 120
```

```
X4 = 5120
```

```
If Text3.Text = Text8.Text And Text3.Text = Str(Universo_entrada(1, 2)) Then
```

```
    Text4.Text = "1"
```

```
End If
```

```
If Text3.Text <> Text8.Text And Text4.Text = "1" Then
```

```
    MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",  
vbExclamation, "Atención"
```

```
    GoTo ignoredatos
```

```
End If
```

```
If Text4.Text = "0" Then
```

```
    Y4 = 3120
```

```
    frminput1.Line17(11).BorderColor = &H8000&
```

```
    frminput1.Line17(11).BorderStyle = 1
```

```
Else
```

```
    If Text4.Text = "1" Then
```

```
        Y4 = 120
```

```
        frminput1.Line17(11).BorderColor = &H0
```

```
        frminput1.Line17(11).BorderStyle = 3
```

```
    Else
```

```
        MsgBox "La ordenada para el punto 4 solo puede tomar valor de cero o uno", vbExclamation,  
"Atención"
```

```
        GoTo ignoredatos
```

```
    End If
```

```
End If
```

```
If X3 > X4 Then
```

```
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,  
"Atención"
```

```
    GoTo ignoredatos
```

```
End If
```

```
End If
```

If opttriangular.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 3120

 X4 = X3

 Y4 = Y3

 frminput1.Line17(12).BorderColor = &H0

End If

If X2 > X3 Then

 MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
 "Atención"

 GoTo ignoredatos

End If

With frminput1

 .Line17(9).X1 = X1

 .Line17(9).Y1 = Y1

 .Line17(9).X2 = X2

 .Line17(9).Y2 = Y2

 .Line17(9).Visible = True

 .Line17(10).X1 = X2

 .Line17(10).Y1 = Y2

 .Line17(10).X2 = X3

 .Line17(10).Y2 = Y3

 .Line17(10).Visible = True

 .Line17(11).X1 = X3

 .Line17(11).Y1 = Y3

 .Line17(11).X2 = X4

 .Line17(11).Y2 = Y4

 .Line17(11).Visible = True

 .Line17(12).X1 = X4

 .Line17(12).Y1 = Y4

 .Line17(12).X2 = 5120

 .Line17(12).Y2 = 3120

```

.Line17(12).Visible = True
End With
'calculo de pendientes
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
Z5 = (Val(Text8.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z5 - Z3
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z5 - Z3))
    End If
    Punto_pendiente(11, 2) = Z5
End If
If opttriangular.Value = True Then
    temporal = Z3 - Z2
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(11, 2) = Z3
End If
Punto_pendiente(11, 1) = Z1
Punto_pendiente(11, 3) = M1
Punto_pendiente(11, 4) = M2

```

```
frmtermino3prop.Hide
```

```
frminput1.Show
```

```
ignoredatos:
```

```
End Sub
```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```
    Label4.Enabled = True
```

```
    Text4.Enabled = True
```

```
    Text8.Enabled = True
```

```
    Label8.Enabled = True
```

```
    Text7.Text = "1"
```

```
    Text8.Text = Universo_entrada(1, 2)
```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```
    Label4.Enabled = False
```

```
    Text4.Enabled = False
```

```
    Text8.Enabled = False
```

```
    Label8.Enabled = False
```

```
    Text7.Text = "0"
```

```
    Text8.Text = Universo_entrada(1, 2)
```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmtermino3prop.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Option Explicit
```

```
Dim temporal, temporal1 As Variant
```

```
Dim Universo, paso As Variant
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Dim Z1, Z2, Z3, Z4, Z5, M1, M2 As Byte
```

Private Sub CancelButton_Click()

```

If Text3.Text <> Text8.Text And Text4.Text = "1" Then
    MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
vbExclamation, "Atención"
    GoTo ignorecancelar
End If
frmtermino3prop2.Hide
ignorecancelar:
End Sub

```

Private Sub Form_Load()

```

Text1.Text = "50"
Text2.Text = "75"
Text3.Text = "100"
Text4.Text = "1"
Text8.Text = Universo_entrada(2, 2)
Text9.Text = frminput2.Lstterminos.List(2)
opttriangular.Value = False
Opttrapezoidal.Value = True
End Sub

```

Private Sub OKButton_Click()

```

Universo = Universo_entrada(2, 2) - Universo_entrada(2, 1)
paso = Universo / 255
frminput2.Lstterminos.List(2) = Text9.Text
If frminput2.Lstterminos.List(2) <> "term 3" Then
    frmtermino3prop2.Caption = "Propiedades de " + Text9.Text
End If
If Opttrapezoidal.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 120
    X4 = 5120
    If Text3.Text = Text8.Text And Text3.Text = Str(Universo_entrada(2, 2)) Then

```

```

    Text4.Text = "1"
End If

If Text3.Text <> Text8.Text And Text4.Text = "1" Then
    MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
vbExclamation, "Atención"
    GoTo ignoredatos
End If
If Text4.Text = "0" Then
    Y4 = 3120
    frminput2.Line17(11).BorderColor = &H8000&
    frminput2.Line17(11).BorderStyle = 1
Else
    If Text4.Text = "1" Then
        Y4 = 120
        frminput2.Line17(11).BorderColor = &H0
        frminput2.Line17(11).BorderStyle = 3
    Else
        MsgBox "La ordenada para el punto 4 solo puede tomar valor de cero o uno", vbExclamation,
"Atención"
        GoTo ignoredatos
    End If
End If
If X3 > X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
"Atención"
    GoTo ignoredatos
End If
End If
If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3

```

```

Y4 = Y3
frminput2.Line17(12).BorderColor = &H0
End If

If X2 > X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

With frminput2
    .Line17(9).X1 = X1
    .Line17(9).Y1 = Y1
    .Line17(9).X2 = X2
    .Line17(9).Y2 = Y2
    .Line17(9).Visible = True
    .Line17(10).X1 = X2
    .Line17(10).Y1 = Y2
    .Line17(10).X2 = X3
    .Line17(10).Y2 = Y3
    .Line17(10).Visible = True
    .Line17(11).X1 = X3
    .Line17(11).Y1 = Y3
    .Line17(11).X2 = X4
    .Line17(11).Y2 = Y4
    .Line17(11).Visible = True
    .Line17(12).X1 = X4
    .Line17(12).Y1 = Y4
    .Line17(12).X2 = 5120
    .Line17(12).Y2 = 3120
    .Line17(12).Visible = True
End With

'calculo de pendientes
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
Z5 = (Val(Text8.Text) / Universo) * 255

```

```

If Opttrapezoidal.Value = True Then
    temporal = Z5 - Z3
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z5 - Z3))
    End If
    Punto_pendiente(12, 2) = Z5
End If
If opttriangular.Value = True Then
    temporal = Z3 - Z2
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(12, 2) = Z3
End If

Punto_pendiente(12, 1) = Z1
Punto_pendiente(12, 3) = M1
Punto_pendiente(12, 4) = M2

frmtermino3prop2.Hide
frminput2.Show
ignoredatos:
End Sub

Private Sub Opttrapezoidal_Click()

```



```

If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text8.Text = Universo_entrada(2, 2)
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
    Text8.Text = Universo_entrada(2, 2)
End If
End Sub

```

```

Private Sub Text9_Change()
frmtermino3prop2.Caption = "Propiedades del término " + Text9.Text
End Sub

```

```

Option Explicit
Dim temporal, temporal1 As Variant
Dim Universo, paso As Variant
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
Dim Z1, Z2, Z3, Z4, Z5, M1, M2 As Byte

```

```

Private Sub CancelButton_Click()
If Text3.Text <> Text8.Text And Text4.Text = "1" Then
    MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
vbExclamation, "Atención"
    GoTo ignorecancelar
End If
frmtermino3prop3.Hide

```

ignorecancelar:

End Sub

Private Sub Form_Load()

Text1.Text = "50"

Text2.Text = "75"

Text3.Text = "100"

Text4.Text = "1"

Text8.Text = Universo_entrada(3, 2)

Text9.Text = frminput3.Lstterminos.List(2)

opttriangular.Value = False

Opttrapezoidal.Value = True

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(3, 2) - Universo_entrada(3, 1)

paso = Universo / 255

frminput3.Lstterminos.List(2) = Text9.Text

If frminput3.Lstterminos.List(2) <> "term 3" Then

 frmtermino3prop3.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120

 X4 = 5120

 If Text3.Text = Text8.Text And Text3.Text = Str(Universo_entrada(3, 2)) Then

 Text4.Text = "1"

 End If

 If Text3.Text <> Text8.Text And Text4.Text = "1" Then

 MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",

vbExclamation, "Atención"

 GoTo ignoredatos

```

End If
If Text4.Text = "0" Then
    Y4 = 3120
    frminput3.Line17(11).BorderColor = &H8000&
    frminput3.Line17(11).BorderStyle = 1
Else
    If Text4.Text = "1" Then
        Y4 = 120
        frminput3.Line17(11).BorderColor = &H0
        frminput3.Line17(11).BorderStyle = 3
    Else
        MsgBox "La ordenada para el punto 4 solo puede tomar valor de cero o uno", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If
If X3 > X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If
If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput3.Line17(12).BorderColor = &H0
End If

If X2 > X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"

```

```

GoTo ignoredatos
End If
With frminput3
.Line17(9).X1 = X1
.Line17(9).Y1 = Y1
.Line17(9).X2 = X2
.Line17(9).Y2 = Y2
.Line17(9).Visible = True
.Line17(10).X1 = X2
.Line17(10).Y1 = Y2
.Line17(10).X2 = X3
.Line17(10).Y2 = Y3
.Line17(10).Visible = True
.Line17(11).X1 = X3
.Line17(11).Y1 = Y3
.Line17(11).X2 = X4
.Line17(11).Y2 = Y4
.Line17(11).Visible = True
.Line17(12).X1 = X4
.Line17(12).Y1 = Y4
.Line17(12).X2 = 5120
.Line17(12).Y2 = 3120
.Line17(12).Visible = True
End With
'calculo de pendientes
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
Z5 = (Val(Text8.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z5 - Z3
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        'temporal1 = temporal / paso

```

```

    'If (temporal1 - Fix(temporal1)) <= 0.5 Then
    '    temporal1 = Fix(temporal1)
    'End If
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z5 - Z3))
End If
Punto_pendiente(13, 2) = Z5
End If
If opttriangular.Value = True Then
    temporal = Z3 - Z2
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(13, 2) = Z3
End If

Punto_pendiente(13, 1) = Z1
Punto_pendiente(13, 3) = M1
Punto_pendiente(13, 4) = M2

frmtermino3prop3.Hide
frminput3.Show
ignoredatos:
End Sub

```

Private Sub Opttrapezoidal_Click()

```

If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text8.Text = Universo_entrada(3, 2)

```

End If

End Sub

Private Sub opttriangular_Click()

If opttriangular.Value = True Then

Label4.Enabled = False

Text4.Enabled = False

Text8.Enabled = False

Label8.Enabled = False

Text7.Text = "0"

Text8.Text = Universo_entrada(3, 2)

End If

End Sub

Private Sub Text9_Change()

frmtermino3prop3.Caption = "Propiedades del término " + Text9.Text

End Sub

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z5, M1, M2 As Byte

Private Sub CancelButton_Click()

If Text3.Text <> Text8.Text And Text4.Text = "1" Then

MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
vbExclamation, "Atención"

GoTo ignorecancelar

End If

frmtermino3prop4.Hide

ignorecancelar:

End Sub

Private Sub Form_Load()

Text1.Text = "50"

Text2.Text = "75"

```
Text3.Text = "100"
```

```
Text4.Text = "1"
```

```
Text8.Text = Universo_entrada(4, 2)
```

```
Text9.Text = frminput4.Lstterminos.List(2)
```

```
opttriangular.Value = False
```

```
Opttrapezoidal.Value = True
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(4, 2) - Universo_entrada(4, 1)
```

```
paso = Universo / 255
```

```
frminput4.Lstterminos.List(2) = Text9.Text
```

```
If frminput4.Lstterminos.List(2) <> "term 3" Then
```

```
    frmtermino3prop4.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
```

```
    Y3 = 120
```

```
    X4 = 5120
```

```
If Text3.Text = Text8.Text And Text3.Text = Str(Universo_entrada(4, 2)) Then
```

```
    Text4.Text = "1"
```

```
End If
```

```
If Text3.Text <> Text8.Text And Text4.Text = "1" Then
```

```
    MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
```

```
vbExclamation, "Atención"
```

```
    GoTo ignoredatos
```

```
End If
```

```
If Text4.Text = "0" Then
```

```
    Y4 = 3120
```

```
    frminput4.Line17(11).BorderColor = &H8000&
```

```
    frminput4.Line17(11).BorderStyle = 1
```

```
Else
```

```

    If Text4.Text = "1" Then
        Y4 = 120
        frminput4.Line17(11).BorderColor = &H0
        frminput4.Line17(11).BorderStyle = 3
    Else
        MsgBox "La ordenada para el punto 4 solo puede tomar valor de cero o uno", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If
If X3 > X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If
If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput4.Line17(12).BorderColor = &H0
End If

If X2 > X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
With frminput4
    .Line17(9).X1 = X1
    .Line17(9).Y1 = Y1
    .Line17(9).X2 = X2

```



```

.Line17(9).Y2 = Y2
.Line17(9).Visible = True
.Line17(10).X1 = X2
.Line17(10).Y1 = Y2
.Line17(10).X2 = X3
.Line17(10).Y2 = Y3
.Line17(10).Visible = True
.Line17(11).X1 = X3
.Line17(11).Y1 = Y3
.Line17(11).X2 = X4
.Line17(11).Y2 = Y4
.Line17(11).Visible = True
.Line17(12).X1 = X4
.Line17(12).Y1 = Y4
.Line17(12).X2 = 5120
.Line17(12).Y2 = 3120
.Line17(12).Visible = True
End With
'calculo de pendientes
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
Z5 = (Val(Text8.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z5 - Z3
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z5 - Z3))
    End If
End If

```

```

    Punto_pendiente(13, 2) = Z5
End If
If opttriangular.Value = True Then
    temporal = Z3 - Z2
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(13, 2) = Z3
End If

```

```

Punto_pendiente(14, 1) = Z1
Punto_pendiente(14, 2) = Z5
Punto_pendiente(14, 3) = M1
Punto_pendiente(14, 4) = M2

```

```

frmtermino3prop4.Hide
frminput4.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text8.Text = Universo_entrada(1, 2)
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then

```

```

Label4.Enabled = False
Text4.Enabled = False
Text8.Enabled = False
Label8.Enabled = False
Text7.Text = "0"
Text8.Text = Universo_entrada(1, 2)
End If
End Sub
Private Sub Text9_Change()
frmtermino3prop4.Caption = "Propiedades del término " + Text9.Text
End Sub

```

```

Option Explicit
Dim temporal, temporal1 As Variant
Dim Universo, paso As Variant
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
Dim Z1, Z2, Z3, Z4, Z5, M1, M2 As Byte

```

```

Private Sub CancelButton_Click()
If Text3.Text <> Text8.Text And Text4.Text = "1" Then
    MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
vbExclamation, "Atención"
    GoTo ignorecancelar
End If
frmtermino3prop5.Hide
ignorecancelar:
End Sub

```

```

Private Sub Form_Load()
Text1.Text = "50"
Text2.Text = "75"
Text3.Text = "100"
Text4.Text = "1"
Text8.Text = Universo_entrada(5, 2)
Text9.Text = frminput5.Lstterminos.List(2)
opttriangular.Value = False
Opttrapezoidal.Value = True

```

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(5, 2) - Universo_entrada(5, 1)

paso = Universo / 255

frminput5.Lstterminos.List(2) = Text9.Text

If frminput5.Lstterminos.List(2) <> "term 3" Then

 frmtermino3prop5.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120

 X4 = 5120

 If Text3.Text = Text8.Text And Text3.Text = Str(Universo_entrada(5, 2)) Then

 Text4.Text = "1"

 End If

 If Text3.Text <> Text8.Text And Text4.Text = "1" Then

 MsgBox "La definición de el término es incorrecta, el valor de la última ordenada debe ser cero",
vbExclamation, "Atención"

 GoTo ignoredatos

 End If

 If Text4.Text = "0" Then

 Y4 = 3120

 frminput5.Line17(11).BorderColor = &H8000&

 frminput5.Line17(11).BorderStyle = 1

 Else

 If Text4.Text = "1" Then

 Y4 = 120

 frminput5.Line17(11).BorderColor = &H0

 frminput5.Line17(11).BorderStyle = 3

 Else

```

        MsgBox "La ordenada para el punto 4 solo puede tomar valor de cero o uno", vbExclamation,
        "Atención"
        GoTo ignoredatos
    End If
End If
If X3 > X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If
If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput5.Line17(12).BorderColor = &H0
End If

If X2 > X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
With frminput5
    .Line17(9).X1 = X1
    .Line17(9).Y1 = Y1
    .Line17(9).X2 = X2
    .Line17(9).Y2 = Y2
    .Line17(9).Visible = True
    .Line17(10).X1 = X2
    .Line17(10).Y1 = Y2
    .Line17(10).X2 = X3

```

```

.Line17(10).Y2 = Y3
.Line17(10).Visible = True
.Line17(11).X1 = X3
.Line17(11).Y1 = Y3
.Line17(11).X2 = X4
.Line17(11).Y2 = Y4
.Line17(11).Visible = True
.Line17(12).X1 = X4
.Line17(12).Y1 = Y4
.Line17(12).X2 = 5120
.Line17(12).Y2 = 3120
.Line17(12).Visible = True
End With
'calculo de pendientes
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
Z5 = (Val(Text8.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    temporal = Z5 - Z3
    If temporal = 0 Then
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        'temporal1 = temporal / paso
        'If (temporal1 - Fix(temporal1)) <= 0.5 Then
        '    temporal1 = Fix(temporal1)
        'End If
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z5 - Z3))
    End If
    Punto_pendiente(15, 2) = Z5
End If
If opttriangular.Value = True Then
    temporal = Z3 - Z2
    If temporal = 0 Then

```

```

        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = &H0
    Else
        M1 = Fix(&HFF / (Z2 - Z1))
        M2 = Fix(&HFF / (Z3 - Z2))
    End If
    Punto_pendiente(15, 2) = Z3
End If
Punto_pendiente(15, 1) = Z1
Punto_pendiente(15, 3) = M1
Punto_pendiente(15, 4) = M2
frmtermino3prop5.Hide
frminput5.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()

```

```

If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text8.Text = Universo_entrada(5, 2)
End If
End Sub

```

```

Private Sub opttriangular_Click()

```

```

If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
    Text8.Text = Universo_entrada(5, 2)
End If
End Sub

```

```
Private Sub Text9_Change()
```

```
frmtermino3prop5.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

```
Option Explicit
```

```
Dim temporal, temporal1 As Variant
```

```
Dim Universo, paso As Variant
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
```

```
frmtermino4prop.Hide
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
Text9.Text = frminput1.Lstterminos.List(3)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "12.5"
```

```
Text2.Text = "25"
```

```
Text3.Text = "37.5"
```

```
Text4.Text = "0"
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(1, 2) - Universo_entrada(1, 1)
```

```
paso = Universo / 255
```

```
frminput1.Lstterminos.List(3) = Text9.Text
```

```
If Text9.Text <> "term 4" Then
```

```
    frmtermino4prop.Caption = "Propiedades del término " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
```

```
    Y1 = 3120
```



```

X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput1.Line17(15).BorderColor = &H0
    frminput1.Line17(15).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(1, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else

```

```

If Opttrapezoidal.Value = True And X4 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(1, 2)),
vbExclamation, "Atención"
    GoTo ignoredatos
End If
End If
With frminput1
    .Line17(13).X1 = X1
    .Line17(13).Y1 = Y1
    .Line17(13).X2 = X2
    .Line17(13).Y2 = Y2
    .Line17(13).Visible = True
    .Line17(14).X1 = X2
    .Line17(14).Y1 = Y2
    .Line17(14).X2 = X3
    .Line17(14).Y2 = Y3
    .Line17(14).Visible = True
    .Line17(15).X1 = X3
    .Line17(15).Y1 = Y3
    .Line17(15).X2 = X4
    .Line17(15).Y2 = Y4
    .Line17(15).Visible = True
    .Line17(16).X1 = X4
    .Line17(16).Y1 = Y4
    .Line17(16).X2 = 5120
    .Line17(16).Y2 = 3120
    .Line17(16).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))

```

```

    Punto_pendiente(16, 2) = Z4
End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z3 - Z2))
    Punto_pendiente(16, 2) = Z3
End If

```

```

Punto_pendiente(16, 1) = Z1
Punto_pendiente(16, 3) = M1
Punto_pendiente(16, 4) = M2

```

```

frmtermino4prop.Hide
frminput1.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    'Text4.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

Private Sub Text9_Change()

frmtermino4prop.Caption = "Propiedades del término " + Text9.Text

End Sub

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub Form_Activate()

Text9.Text = frminput2.Lstterminos.List(3)

End Sub

Private Sub Form_Load()

Text1.Text = "12.5"

Text2.Text = "25"

Text3.Text = "37.5"

Text4.Text = "0"

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(2, 2) - Universo_entrada(2, 1)

paso = Universo / 255

frminput2.Lstterminos.List(3) = Text9.Text

If frminput2.Lstterminos.List(3) <> "term 4" Then

 frmtermino4prop2.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120

```

X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

```

```

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput2.Line17(15).BorderColor = &H0
    frminput2.Line17(15).BorderColor = &H0
End If

```

```

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(2, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(2, 2)),
        vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

```

```

    End If
End If
With frminput2
    .Line17(13).X1 = X1
    .Line17(13).Y1 = Y1
    .Line17(13).X2 = X2
    .Line17(13).Y2 = Y2
    .Line17(13).Visible = True
    .Line17(14).X1 = X2
    .Line17(14).Y1 = Y2
    .Line17(14).X2 = X3
    .Line17(14).Y2 = Y3
    .Line17(14).Visible = True
    .Line17(15).X1 = X3
    .Line17(15).Y1 = Y3
    .Line17(15).X2 = X4
    .Line17(15).Y2 = Y4
    .Line17(15).Visible = True
    .Line17(16).X1 = X4
    .Line17(16).Y1 = Y4
    .Line17(16).X2 = 5120
    .Line17(16).Y2 = 3120
    .Line17(16).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(17, 2) = Z4
End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))

```

```

M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(17, 2) = Z3
End If

```

```

Punto_pendiente(17, 1) = Z1
Punto_pendiente(17, 3) = M1
Punto_pendiente(17, 4) = M2

```

```

frmtermino4prop2.Hide
frminput2.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    'Text4.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```

Private Sub Text9_Change()
frmtermino4prop2.Caption = "Propiedades del término " + Text9.Text
End Sub

```

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino4prop3.Hide

End Sub

Private Sub Form_Activate()

Text9.Text = frminput3.Lstterminos.List(3)

End Sub

Private Sub Form_Load()

Text1.Text = "12.5"

Text2.Text = "25"

Text3.Text = "37.5"

Text4.Text = "0"

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(3, 2) - Universo_entrada(3, 1)

paso = Universo / 255

frminput3.Lstterminos.List(3) = Text9.Text

If frminput3.Lstterminos.List(3) <> "term 4" Then

 frmtermino4prop3.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120


```

X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

```

```

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput3.Line17(15).BorderColor = &H0
    frminput3.Line17(15).BorderColor = &H0
End If

```

```

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

```

```

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(3, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(3, 2)),
        vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

```

```

    End If
End If
With frminput3
    .Line17(13).X1 = X1
    .Line17(13).Y1 = Y1
    .Line17(13).X2 = X2
    .Line17(13).Y2 = Y2
    .Line17(13).Visible = True
    .Line17(14).X1 = X2
    .Line17(14).Y1 = Y2
    .Line17(14).X2 = X3
    .Line17(14).Y2 = Y3
    .Line17(14).Visible = True
    .Line17(15).X1 = X3
    .Line17(15).Y1 = Y3
    .Line17(15).X2 = X4
    .Line17(15).Y2 = Y4
    .Line17(15).Visible = True
    .Line17(16).X1 = X4
    .Line17(16).Y1 = Y4
    .Line17(16).X2 = 5120
    .Line17(16).Y2 = 3120
    .Line17(16).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(18, 2) = Z4
End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))

```

```

M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(18, 2) = Z3
End If

```

```

Punto_pendiente(18, 1) = Z1
Punto_pendiente(18, 3) = M1
Punto_pendiente(18, 4) = M2

```

```

frmtermino4prop3.Hide
frminput3.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    'Text4.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```

End Sub
Private Sub Text9_Change()
frmtermino4prop3.Caption = "Propiedades del término " + Text9.Text
End Sub

```

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino4prop4.Hide

End Sub

Private Sub Form_Activate()

Text9.Text = frminput4.Lstterminos.List(3)

End Sub

Private Sub Form_Load()

Text1.Text = "12.5"

Text2.Text = "25"

Text3.Text = "37.5"

Text4.Text = "0"

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(4, 2) - Universo_entrada(4, 1)

paso = Universo / 255

frminput4.Lstterminos.List(3) = Text9.Text

If frminput4.Lstterminos.List(3) <> "term 3" Then

 frmtermino4prop4.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120

 X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)

```

Y4 = 3120
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

```

```

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput4.Line17(15).BorderColor = &H0
    frminput4.Line17(15).BorderColor = &H0
End If

```

```

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
        vbExclamation, "Atención"
        GoTo ignoredatos
    End If

```

End If

With frminput4

```
.Line17(13).X1 = X1
.Line17(13).Y1 = Y1
.Line17(13).X2 = X2
.Line17(13).Y2 = Y2
.Line17(13).Visible = True
.Line17(14).X1 = X2
.Line17(14).Y1 = Y2
.Line17(14).X2 = X3
.Line17(14).Y2 = Y3
.Line17(14).Visible = True
.Line17(15).X1 = X3
.Line17(15).Y1 = Y3
.Line17(15).X2 = X4
.Line17(15).Y2 = Y4
.Line17(15).Visible = True
.Line17(16).X1 = X4
.Line17(16).Y1 = Y4
.Line17(16).X2 = 5120
.Line17(16).Y2 = 3120
.Line17(16).Visible = True
```

End With

'calculo de pendientes

```
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
```

If Opttrapezoidal.Value = True Then

```
M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z4 - Z3))
Punto_pendiente(19, 2) = Z4
```

End If

If opttriangular.Value = True Then

```
M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z3 - Z2))
```

```
Punto_pendiente(19, 2) = Z3
End If
```

```
Punto_pendiente(19, 1) = Z1
Punto_pendiente(19, 3) = M1
Punto_pendiente(19, 4) = M2
```

```
frmtermino4prop4.Hide
frminput4.Show
ignoredatos:
End Sub
```

```
Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    'Text4.Locked = False
End If
End Sub
```

```
Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub
```

```
Private Sub Text9_Change()
frmtermino4prop4.Caption = "Propiedades del término " + Text9.Text
End Sub
```

```
Option Explicit
```

```
Dim temporal, temporal1 As Variant
Dim Universo, paso As Variant
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
```

```
frmtermino4prop5.Hide
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
Text9.Text = frminput5.Lstterminos.List(3)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "12.5"
```

```
Text2.Text = "25"
```

```
Text3.Text = "37.5"
```

```
Text4.Text = "0"
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(5, 2) - Universo_entrada(5, 1)
```

```
paso = Universo / 255
```

```
frminput5.Lstterminos.List(3) = Text9.Text
```

```
If frminput5.Lstterminos.List(3) <> "term 4" Then
```

```
    frmtermino4prop5.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
```

```
    Y3 = 120
```

```
    X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
```

```
    Y4 = 3120
```



```

If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

```

```

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput5.Line17(15).BorderColor = &H0
    frminput5.Line17(15).BorderColor = &H0
End If

```

```

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

```

```

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
        vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

```

With frminput5

```
.Line17(13).X1 = X1
.Line17(13).Y1 = Y1
.Line17(13).X2 = X2
.Line17(13).Y2 = Y2
.Line17(13).Visible = True
.Line17(14).X1 = X2
.Line17(14).Y1 = Y2
.Line17(14).X2 = X3
.Line17(14).Y2 = Y3
.Line17(14).Visible = True
.Line17(15).X1 = X3
.Line17(15).Y1 = Y3
.Line17(15).X2 = X4
.Line17(15).Y2 = Y4
.Line17(15).Visible = True
.Line17(16).X1 = X4
.Line17(16).Y1 = Y4
.Line17(16).X2 = 5120
.Line17(16).Y2 = 3120
.Line17(16).Visible = True
```

End With

'calculo de pendientes

```
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
```

If Opttrapezoidal.Value = True Then

```
M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z4 - Z3))
Punto_pendiente(20, 2) = Z4
```

End If

If opttriangular.Value = True Then

```
M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(20, 2) = Z3
```

End If

Punto_pendiente(20, 1) = Z1

Punto_pendiente(20, 3) = M1

Punto_pendiente(20, 4) = M2

frmtermino4prop5.Hide

frminput5.Show

ignoredatos:

End Sub

Private Sub Opttrapezoidal_Click()

If Opttrapezoidal.Value = True Then

Label4.Enabled = True

Text4.Enabled = True

Text8.Enabled = True

Label8.Enabled = True

Text7.Text = "1"

Text4.Locked = False

End If

End Sub

Private Sub opttriangular_Click()

If opttriangular.Value = True Then

Label4.Enabled = False

Text4.Enabled = False

Text8.Enabled = False

Label8.Enabled = False

Text7.Text = "0"

End If

End Sub

Private Sub Text9_Change()

frmtermino4prop5.Caption = "Propiedades del término " + Text9.Text

End Sub

Option Explicit

```
Dim temporal, temporal1 As Variant
Dim Universo, paso As Variant
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
```

```
frmtermino5prop.Hide
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
Text9.Text = frminput1.Lstterminos.List(4)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "62.5"
```

```
Text2.Text = "75"
```

```
Text3.Text = "87.5"
```

```
Text4.Text = "0"
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(1, 2) - Universo_entrada(1, 1)
```

```
paso = Universo / 255
```

```
frminput1.Lstterminos.List(4) = Text9.Text
```

```
If Text9.Text <> "term 5" Then
```

```
    frmtermino5prop.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```
    Y2 = 120
```

```
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
```

```
    Y3 = 120
```

```
    X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
```

```
    Y4 = 3120
```

```

frminput1.Line17(19).BorderColor = &HC0C0&
frminput1.Line17(20).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

```

```

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput1.Line17(19).BorderColor = &H0
    frminput1.Line17(20).BorderColor = &H0
End If

```

```

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

```

```

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(1, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(1, 2)),
        vbExclamation, "Atención"
        GoTo ignoredatos
    End If
End If

```

```

End If
End If
With frminput1
.Line17(17).X1 = X1
.Line17(17).Y1 = Y1
.Line17(17).X2 = X2
.Line17(17).Y2 = Y2
.Line17(17).Visible = True
.Line17(18).X1 = X2
.Line17(18).Y1 = Y2
.Line17(18).X2 = X3
.Line17(18).Y2 = Y3
.Line17(18).Visible = True
.Line17(19).X1 = X3
.Line17(19).Y1 = Y3
.Line17(19).X2 = X4
.Line17(19).Y2 = Y4
.Line17(19).Visible = True
.Line17(20).X1 = X4
.Line17(20).Y1 = Y4
.Line17(20).X2 = 5120
.Line17(20).Y2 = 3120
.Line17(20).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z4 - Z3))
Punto_pendiente(21, 2) = Z4
End If
If opttriangular.Value = True Then
M1 = Fix(&HFF / (Z2 - Z1))

```

```

M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(21, 2) = Z3
End If

```

```

Punto_pendiente(21, 1) = Z1
Punto_pendiente(21, 3) = M1
Punto_pendiente(21, 4) = M2

```

```

frmtermino5prop.Hide
frminput1.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    'txtpuntocuatro2.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```

End Sub
Private Sub Text9_Change()
frmtermino5prop.Caption = "Propiedades del término " + Text9.Text
End Sub

```

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino5prop2.Hide

End Sub

Private Sub Form_Activate()

Text9.Text = frminput2.Lstterminos.List(4)

End Sub

Private Sub Form_Load()

Text1.Text = "62.5"

Text2.Text = "75"

Text3.Text = "87.5"

Text4.Text = "0"

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(2, 2) - Universo_entrada(2, 1)

paso = Universo / 255

frminput2.Lstterminos.List(4) = Text9.Text

If frminput2.Lstterminos.List(4) <> "term 5" Then

 frmtermino5prop2.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120

 X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)


```

Y4 = 3120
frminput2.Line17(19).BorderColor = &HC0C0&
frminput2.Line17(20).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput2.Line17(19).BorderColor = &H0
    frminput2.Line17(20).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(2, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(2, 2)),
        vbExclamation, "Atención"
    
```

```

        GoTo ignoredatos
    End If
End If
With frminput2
    .Line17(17).X1 = X1
    .Line17(17).Y1 = Y1
    .Line17(17).X2 = X2
    .Line17(17).Y2 = Y2
    .Line17(17).Visible = True
    .Line17(18).X1 = X2
    .Line17(18).Y1 = Y2
    .Line17(18).X2 = X3
    .Line17(18).Y2 = Y3
    .Line17(18).Visible = True
    .Line17(19).X1 = X3
    .Line17(19).Y1 = Y3
    .Line17(19).X2 = X4
    .Line17(19).Y2 = Y4
    .Line17(19).Visible = True
    .Line17(20).X1 = X4
    .Line17(20).Y1 = Y4
    .Line17(20).X2 = 5120
    .Line17(20).Y2 = 3120
    .Line17(20).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(22, 2) = Z4
End If
If opttriangular.Value = True Then

```

```

M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(22, 2) = Z3
End If

```

```

Punto_pendiente(22, 1) = Z1
Punto_pendiente(22, 3) = M1
Punto_pendiente(22, 4) = M2

```

```

frmtermino5prop2.Hide
frminput2.Show
ignoredatos:

```

```
End Sub
```

```
Private Sub Opttrapezoidal_Click()
```

```
If Opttrapezoidal.Value = True Then
```

```

    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    'Text4.Locked = False

```

```
End If
```

```
End Sub
```

```
Private Sub opttriangular_Click()
```

```
If opttriangular.Value = True Then
```

```

    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"

```

```
End If
```

```
End Sub
```

```
Private Sub Text9_Change()
```

```
frmtermino5prop2.Caption = "Propiedades del término " + Text9.Text
```

```
End Sub
```

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino5prop3.Hide

End Sub

Private Sub Form_Activate()

Text9.Text = frminput3.Lstterminos.List(4)

End Sub

Private Sub Form_Load()

text1.Text = "62.5"

text2.Text = "75"

text3.Text = "87.5"

text4.Text = "0"

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(3, 2) - Universo_entrada(3, 1)

paso = Universo / 255

frminput3.Lstterminos.List(4) = Text9.Text

If frminput3.Lstterminos.List(4) <> "term 5" Then

 frmtermino5prop3.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(text3.Text)) * (5000 / Universo)

 Y3 = 120

 X4 = 120 + CInt(Val(text4.Text)) * (5000 / Universo)

```

Y4 = 3120
frminput3.Line17(19).BorderColor = &HC0C0C&
frminput3.Line17(20).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput3.Line17(19).BorderColor = &H0
    frminput3.Line17(20).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(3, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then
        MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(3, 2)),
        vbExclamation, "Atención"
    
```

```

        GoTo ignoredatos
    End If
End If
With frminput3
    .Line17(17).X1 = X1
    .Line17(17).Y1 = Y1
    .Line17(17).X2 = X2
    .Line17(17).Y2 = Y2
    .Line17(17).Visible = True
    .Line17(18).X1 = X2
    .Line17(18).Y1 = Y2
    .Line17(18).X2 = X3
    .Line17(18).Y2 = Y3
    .Line17(18).Visible = True
    .Line17(19).X1 = X3
    .Line17(19).Y1 = Y3
    .Line17(19).X2 = X4
    .Line17(19).Y2 = Y4
    .Line17(19).Visible = True
    .Line17(20).X1 = X4
    .Line17(20).Y1 = Y4
    .Line17(20).X2 = 5120
    .Line17(20).Y2 = 3120
    .Line17(20).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(text1.Text) / Universo) * 255
Z2 = (Val(text2.Text) / Universo) * 255
Z3 = (Val(text3.Text) / Universo) * 255
Z4 = (Val(text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(23, 2) = Z4
End If
If opttriangular.Value = True Then

```

```

M1 = Fix(&HFF / (Z2 - Z1))
M2 = Fix(&HFF / (Z3 - Z2))
Punto_pendiente(23, 2) = Z3
End If

```

```

Punto_pendiente(23, 1) = Z1
Punto_pendiente(23, 3) = M1
Punto_pendiente(23, 4) = M2

```

```

frmtermino5prop3.Hide
frminput3.Show
ignoredatos:
End Sub

```

```
Private Sub Opttrapezoidal_Click()
```

```

If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    'Text4.Locked = False
End If
End Sub

```

```
Private Sub opttriangular_Click()
```

```

If opttriangular.Value = True Then
    Label4.Enabled = False
    text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub

```

```
Private Sub Text9_Change()
```

```

frmtermino5prop3.Caption = "Propiedades del término " + Text9.Text
End Sub

```

Option Explicit

Dim temporal, temporal1 As Variant

Dim Universo, paso As Variant

Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer

Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte

Private Sub CancelButton_Click()

frmtermino5prop4.Hide

End Sub

Private Sub Form_Activate()

Text9.Text = frminput4.Lstterminos.List(4)

End Sub

Private Sub Form_Load()

Text1.Text = "62.5"

Text2.Text = "75"

Text3.Text = "87.5"

Text4.Text = "0"

opttriangular.Value = True

Opttrapezoidal.Value = False

End Sub

Private Sub OKButton_Click()

Universo = Universo_entrada(4, 2) - Universo_entrada(4, 1)

paso = Universo / 255

frminput4.Lstterminos.List(4) = Text9.Text

If frminput4.Lstterminos.List(4) <> "term 5" Then

 frmtermino5prop4.Caption = "Propiedades de " + Text9.Text

End If

If Opttrapezoidal.Value = True Then

 X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)

 Y1 = 3120

 X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)

 Y2 = 120

 X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)

 Y3 = 120


```

X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
frminput4.Line17(19).BorderColor = &HC0C0C&
frminput4.Line17(20).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput4.Line17(19).BorderColor = &H0
    frminput4.Line17(20).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If

If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos
Else
    If Opttrapezoidal.Value = True And X4 > 5120 Then

```

```

    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(4, 2)),
vbExclamation, "Atención"
    GoTo ignoredatos
End If
End If
With frminput4
    .Line17(17).X1 = X1
    .Line17(17).Y1 = Y1
    .Line17(17).X2 = X2
    .Line17(17).Y2 = Y2
    .Line17(17).Visible = True
    .Line17(18).X1 = X2
    .Line17(18).Y1 = Y2
    .Line17(18).X2 = X3
    .Line17(18).Y2 = Y3
    .Line17(18).Visible = True
    .Line17(19).X1 = X3
    .Line17(19).Y1 = Y3
    .Line17(19).X2 = X4
    .Line17(19).Y2 = Y4
    .Line17(19).Visible = True
    .Line17(20).X1 = X4
    .Line17(20).Y1 = Y4
    .Line17(20).X2 = 5120
    .Line17(20).Y2 = 3120
    .Line17(20).Visible = True
End With
'calculo de pendientes
Z0 = (Val(Text5.Text) / Universo) * 255
Z1 = (Val(Text1.Text) / Universo) * 255
Z2 = (Val(Text2.Text) / Universo) * 255
Z3 = (Val(Text3.Text) / Universo) * 255
Z4 = (Val(Text4.Text) / Universo) * 255
If Opttrapezoidal.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(24, 2) = Z4

```

```

End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z3 - Z2))
    Punto_pendiente(24, 2) = Z3
End If

```

```

Punto_pendiente(24, 1) = Z1
Punto_pendiente(24, 3) = M1
Punto_pendiente(24, 4) = M2

```

```

frmtermino5prop4.Hide
frminput4.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text4.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If
End Sub
Private Sub Text9_Change()

```

```
frmtermino5prop4.Caption = "Propiedades del término " + Text9.Text
End Sub
```

```
Option Explicit
```

```
Dim temporal, temporal1 As Variant
```

```
Dim Universo, paso As Variant
```

```
Dim X1, X2, X3, X4, Y1, Y2, Y3, Y4 As Integer
```

```
Dim Z1, Z2, Z3, Z4, Z0, M1, M2 As Byte
```

```
Private Sub CancelButton_Click()
```

```
frmtermino5prop5.Hide
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
Text9.Text = frminput5.Lstterminos.List(4)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Text1.Text = "62.5"
```

```
Text2.Text = "75"
```

```
Text3.Text = "87.5"
```

```
Text4.Text = "0"
```

```
opttriangular.Value = True
```

```
Opttrapezoidal.Value = False
```

```
End Sub
```

```
Private Sub OKButton_Click()
```

```
Universo = Universo_entrada(5, 2) - Universo_entrada(5, 1)
```

```
paso = Universo / 255
```

```
frminput5.Lstterminos.List(4) = Text9.Text
```

```
If frminput5.Lstterminos.List(4) <> "term 5" Then
```

```
    frmtermino5prop5.Caption = "Propiedades de " + Text9.Text
```

```
End If
```

```
If Opttrapezoidal.Value = True Then
```

```
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
```

```
    Y1 = 3120
```

```
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
```

```

Y2 = 120
X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
Y3 = 120
X4 = 120 + CInt(Val(Text4.Text)) * (5000 / Universo)
Y4 = 3120
frminput5.Line17(19).BorderColor = &HC0C0&
frminput5.Line17(20).BorderColor = &H0
If X3 >= X4 Then
    MsgBox "La abscisa del punto 3 no puede ser mayor o igual a la del punto 4", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
End If

If opttriangular.Value = True Then
    X1 = 120 + CInt(Val(Text1.Text)) * (5000 / Universo)
    Y1 = 3120
    X2 = 120 + CInt(Val(Text2.Text)) * (5000 / Universo)
    Y2 = 120
    X3 = 120 + CInt(Val(Text3.Text)) * (5000 / Universo)
    Y3 = 3120
    X4 = X3
    Y4 = Y3
    frminput5.Line17(19).BorderColor = &H0
    frminput5.Line17(20).BorderColor = &H0
End If

If X2 >= X3 Then
    MsgBox "La abscisa del punto 2 no puede ser mayor o igual a la del punto 3", vbExclamation,
    "Atención"
    GoTo ignoredatos
End If
If opttriangular.Value = True And X3 > 5120 Then
    MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(5, 2)),
    vbExclamation, "Atención"
    GoTo ignoredatos

```

Else

If Opttrapezoidal.Value = True And X4 > 5120 Then

MsgBox "Los valores de las abscisas no pueden excederse de " + Str(Universo_entrada(5, 2)),
vbExclamation, "Atención"

GoTo ignoredatos

End If

End If

With frminput5

.Line17(17).X1 = X1

.Line17(17).Y1 = Y1

.Line17(17).X2 = X2

.Line17(17).Y2 = Y2

.Line17(17).Visible = True

.Line17(18).X1 = X2

.Line17(18).Y1 = Y2

.Line17(18).X2 = X3

.Line17(18).Y2 = Y3

.Line17(18).Visible = True

.Line17(19).X1 = X3

.Line17(19).Y1 = Y3

.Line17(19).X2 = X4

.Line17(19).Y2 = Y4

.Line17(19).Visible = True

.Line17(20).X1 = X4

.Line17(20).Y1 = Y4

.Line17(20).X2 = 5120

.Line17(20).Y2 = 3120

.Line17(20).Visible = True

End With

'calculo de pendientes

Z0 = (Val(Text5.Text) / Universo) * 255

Z1 = (Val(Text1.Text) / Universo) * 255

Z2 = (Val(Text2.Text) / Universo) * 255

Z3 = (Val(Text3.Text) / Universo) * 255

Z4 = (Val(Text4.Text) / Universo) * 255

If Opttrapezoidal.Value = True Then

M1 = Fix(&HFF / (Z2 - Z1))

```

    M2 = Fix(&HFF / (Z4 - Z3))
    Punto_pendiente(25, 2) = Z4
End If
If opttriangular.Value = True Then
    M1 = Fix(&HFF / (Z2 - Z1))
    M2 = Fix(&HFF / (Z3 - Z2))
    Punto_pendiente(25, 2) = Z3
End If

```

```

Punto_pendiente(25, 1) = Z1
Punto_pendiente(25, 3) = M1
Punto_pendiente(25, 4) = M2

```

```

frmtermino5prop5.Hide
frminput5.Show
ignoredatos:
End Sub

```

```

Private Sub Opttrapezoidal_Click()
If Opttrapezoidal.Value = True Then
    Label4.Enabled = True
    Text4.Enabled = True
    Text8.Enabled = True
    Label8.Enabled = True
    Text7.Text = "1"
    Text4.Locked = False
End If
End Sub

```

```

Private Sub opttriangular_Click()
If opttriangular.Value = True Then
    Label4.Enabled = False
    Text4.Enabled = False
    Text8.Enabled = False
    Label8.Enabled = False
    Text7.Text = "0"
End If

```

End Sub

Private Sub Text9_Change()

frmtermino5prop5.Caption = "Propiedades del término " + Text9.Text

End Sub

Dim i, j As Integer

Dim indice As Integer

Dim indice2 As Integer

Private Sub Fileexit_Click(Index As Integer)

End

End Sub

Private Sub fileopen_Click(Index As Integer)

dlgbase.CancelError = True

On Error GoTo ManejoError:

dlgbase.Flags = cd10FNHideReadOnly

dlgbase.Filter = "Todos los archivos (*.*)|*. *|archivos ASM (*.asm)|*.asm"

dlgbase.FilterIndex = 2

dlgbase.ShowOpen

Print "Se selecciono:" & dlgbase.FileTitle

Exit Sub

ManejoError:

MsgBox "Se eligió cancelar"

End Sub

Private Sub Form_Load()

For j = 1 To 6

For i = 1 To 5

matriz_indices(i, j) = 0

Next i

Next j

indice2 = 0

input_index = 0

rb_index = 6

output_index = 9

'FileSystem.ChDir ("C:\Documents and Settings\Khalena\Mis documentos\Tesis\MGTEK")

'Set Toolbar1.ImageList = ImageList1

'With Toolbar1.Buttons


```

' .Add 1, "salida", , tbrDefault, "salida"
' .Add 1, "entrada", , tbrDefault, "entrada"
' .Add 1, "bloquederegla", , tbrDefault, "bloque"
' .Item(1).ToolTipText = "Agregar bloque de reglas"
' .Item(2).ToolTipText = "Agregar entrada"
' .Item(3).ToolTipText = "Agregar salida"
'End With
End Sub
Private Sub Frame1_DragDrop(Source As Control, x As Single, y As Single)
If indice = 1 Then
    Picture1(1).Left = x
    Picture1(1).Top = y
    Line1(indice).X1 = Picture1(indice).Left + Picture1(indice).Width
    Line1(indice).X2 = Picture1(7).Left
    Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
    Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If
If indice = 2 Then
    Picture1(2).Left = x
    Picture1(2).Top = y
    Line1(indice).X1 = Picture1(indice).Left + Picture1(indice).Width
    Line1(indice).X2 = Picture1(7).Left
    Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
    Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If
If indice = 3 Then
    Picture1(3).Left = x
    Picture1(3).Top = y
    Line1(indice).X1 = Picture1(indice).Left + Picture1(indice).Width
    Line1(indice).X2 = Picture1(7).Left
    Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
    Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If
If indice = 4 Then
    Picture1(4).Left = x
    Picture1(4).Top = y
    Line1(indice).X1 = Picture1(indice).Left + Picture1(indice).Width

```

```

Line1(indice).X2 = Picture1(7).Left
Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If

If indice = 5 Then
    Picture1(5).Left = x
    Picture1(5).Top = y
    Line1(indice).X1 = Picture1(indice).Left + Picture1(indice).Width
    Line1(indice).X2 = Picture1(7).Left
    Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
    Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If

If indice = 7 Then
    Picture1(7).Left = x
    Picture1(7).Top = y
    For lazo = 1 To 5
        If Line1(lazo).Visible = True Then
            Line1(lazo).X1 = Picture1(lazo).Left + Picture1(lazo).Width
            Line1(lazo).X2 = Picture1(7).Left
            Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)
            Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
        End If
    Next lazo
    For lazo = 10 To 12
        If Line1(lazo).Visible = True Then
            Line1(lazo).X1 = Picture1(lazo).Left
            Line1(lazo).X2 = Picture1(7).Left + Picture1(7).Width
            Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)
            Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
        End If
    Next lazo
End If

If indice = 8 Then
    Picture1(8).Left = x
    Picture1(8).Top = y
    ' Line1(indice).X1 = Picture1(indice).Left
    ' Line1(indice).X2 = Picture1(7).Left + Picture1(7).Width

```

```

' Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
' Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If
If indice = 10 Then
    Picture1(10).Left = x
    Picture1(10).Top = y
    Line1(indice).X1 = Picture1(indice).Left
    Line1(indice).X2 = Picture1(7).Left + Picture1(7).Width
    Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
    Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If
If indice = 11 Then
    Picture1(11).Left = x
    Picture1(11).Top = y
    Line1(indice).X1 = Picture1(indice).Left
    Line1(indice).X2 = Picture1(7).Left + Picture1(7).Width
    Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
    Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If
If indice = 12 Then
    Picture1(12).Left = x
    Picture1(12).Top = y
    Line1(indice).X1 = Picture1(indice).Left
    Line1(indice).X2 = Picture1(7).Left + Picture1(7).Width
    Line1(indice).Y1 = Picture1(indice).Top + Round((Picture1(indice).Height / 2), 0)
    Line1(indice).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
End If
End Sub

```

```
Private Sub Label3_DblClick()
```

```
ee.Show
```

```
End Sub
```

```
Private Sub newfile_Click(Index As Integer)
```

```
ee.Show
```

```
'ee.Text1.Text = ee.txtinput + ee.txtoutput + ee.txtruleblock
```

End Sub

Private Sub OLE1_DragOver(Source As Control, x As Single, y As Single, State As Integer)

lblmensaje.Caption = " X: " & x & " Y: " & y

End Sub

Private Sub Picture1_DragOver(Index As Integer, Source As Control, x As Single, y As Single, State As Integer)

indice = Index

End Sub

Private Sub Picture1_MouseDown(Index As Integer, Button As Integer, Shift As Integer, x As Single, y As Single)

If Index = 1 Then

 frminput1.Show

End If

If Index = 2 Then

 frminput2.Show

End If

If Index = 3 Then

 frminput3.Show

End If

If Index = 4 Then

 frminput4.Show

End If

If Index = 5 Then

 frminput5.Show

End If

'aqui empieza el invento

If Index = 7 Then

 With frmruleblock1

 .Frame1.Height = 315 * input_index + 368 * (input_index + 1)

 .Frame1.Top = (.Frame2.Top - .Frame1.Height) / 2

 For lazo = 1 To input_index

 .Combo1(lazo).Top = .Frame1.Top + 368 * (lazo) + .Combo1(lazo).Height * (lazo - 1)

 .Combo1(lazo).Left = 480

 .Combo1(lazo).Visible = True

```

        .Text1(lazo).Text = VarNames(lazo)
        .Text1(lazo).Top = .Combo1(lazo).Top - .Text1(lazo).Height
        .Text1(lazo).Left = 480
        .Text1(lazo).Visible = True
    Next lazo
    For lazo = 1 To output_index - 9
        .Frame2.Width = .Combo6(1).Width * lazo + 240 * (lazo + 1)
        .Combo6(lazo).Top = 4920
        .Combo6(lazo).Left = .Frame2.Left + 240 * (lazo) + .Combo6(1).Width * (lazo - 1)
        .Combo6(lazo).Visible = True
        .Text6(lazo).Text = OutNames(lazo)
        .Text6(lazo).Top = .Combo6(1).Top - .Text6(1).Height
        .Text6(lazo).Left = .Combo6(lazo).Left
        .Text6(lazo).Visible = True
    Next lazo
    .Show
End With
End If
'fin del mundo

If Index = 10 Then
    frmoutput1.Show
End If
If Index = 11 Then
    frmoutput2.Show
End If
If Index = 12 Then
    frmoutput3.Show
End If
datosenmatriz:
End Sub

Public Sub MATRIZ()
    Dim index_list(1 To 5) As Variant
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)

```

Select Case Button.Key

Case "entrada"

If rb_index < 7 Then

MsgBox "Debe agregar al menos un bloque de reglas", vbExclamation, "Atención"

GoTo Ignoreentrada

End If

input_index = input_index + 1

If input_index > 5 Then

input_index = 5

MsgBox "Se excedió número máximo de entradas", vbExclamation, "Atención"

GoTo Ignoreentrada

End If

For lazo = 1 To input_index

Picture1(lazo).Left = 886

Picture1(lazo).Top = (lazo - 1) * Picture1(1).Height + lazo * (Frame1.Height - input_index * (Picture1(1).Height)) / (input_index + 1)

Next lazo

If rb_index = 7 Then

For lazo = 1 To input_index

Line1(lazo).X1 = Picture1(lazo).Left + Picture1(lazo).Width

Line1(lazo).X2 = Picture1(7).Left

Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)

Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)

Line1(lazo).Visible = True

Next lazo

Else

For lazo = 1 To input_index

If Line1(lazo).Visible = True Then

Line1(lazo).X1 = Picture1(lazo).Left + Picture1(lazo).Width

Line1(lazo).X2 = Picture1(7).Left

Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)

Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)

End If

Next

End If

Picture1(input_index).Picture = LoadPicture("C:\Archivos de programa\Microsoft Visual Studio\VB98\input.bmp")

```

matriz_indices(input_index, 1) = input_index
matriz_indices(input_index, 2) = 3
VarNames(input_index) = "Variable " + Str(input_index)
If input_index = 1 Then
    For lazo = 1 To matriz_indices(1, 2)
        frminput1.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo1(input_index).AddItem "term" + Str(lazo)
    Next lazo
End If
If input_index = 2 Then
    For lazo = 1 To matriz_indices(2, 2)
        frminput2.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo1(input_index).AddItem "term" + Str(lazo)
    Next lazo
End If
If input_index = 3 Then
    For lazo = 1 To matriz_indices(3, 2)
        frminput3.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo1(input_index).AddItem "term" + Str(lazo)
    Next lazo
End If
If input_index = 4 Then
    For lazo = 1 To matriz_indices(4, 2)
        frminput4.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo1(input_index).AddItem "term" + Str(lazo)
    Next lazo
End If
If input_index = 5 Then
    For lazo = 1 To matriz_indices(5, 2)
        frminput5.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo1(input_index).AddItem "term" + Str(lazo)
    Next lazo
End If

Case "salida"
    If rb_index < 7 Then

```

```

    MsgBox "Debe agregar al menos un bloque de reglas", vbExclamation, "Atención"
    GoTo Ignoresalida
End If
If input_index < 1 Then
    MsgBox "El sistema necesita al menos una entrada", vbExclamation, "Atención"
    GoTo Ignoresalida
End If
output_index = output_index + 1
If output_index > 12 Then
    output_index = 12
    MsgBox "Se excedió número máximo de salidas", vbExclamation, "Atención"
    GoTo Ignoresalida
End If
For lazo = 10 To output_index
    Picture1(lazo).Left = 6565
    Picture1(lazo).Top = (lazo - 10) * Picture1(10).Height + (lazo - 9) * (Frame1.Height -
(output_index - 9) * (Picture1(10).Height)) / (output_index - 8)
Next lazo
If rb_index = 7 Then
    For lazo = 10 To output_index
        Line1(lazo).X1 = Picture1(lazo).Left
        Line1(lazo).X2 = Picture1(7).Left + Picture1(7).Width
        Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)
        Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
        Line1(lazo).Visible = True
    Next lazo
Else
    For lazo = 10 To output_index
        Line1(lazo).X1 = Picture1(lazo).Left
        Line1(lazo).X2 = Picture1(7).Left + Picture1(7).Width
        Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)
        Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
    Next lazo
End If
Picture1(output_index).Picture = LoadPicture("C:\Archivos de programa\Microsoft Visual Studio\VB98\
output.bmp")
matriz_indices(output_index - 9, 3) = output_index - 9

```



```

matriz_indices(output_index - 9, 4) = 3
OutNames(output_index - 9) = "Salida " + Str(output_index - 9)
If output_index = 10 Then
    For lazo = 1 To matriz_indices(1, 4)
        frmoutput1.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo6(1).AddItem "term" + Str(lazo)
    Next lazo
End If
If output_index = 11 Then
    For lazo = 1 To matriz_indices(2, 4)
        frmoutput2.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo6(2).AddItem "term" + Str(lazo)
    Next lazo
End If
If output_index = 12 Then
    For lazo = 1 To matriz_indices(3, 4)
        frmoutput3.Lstterminos.AddItem "term" + Str(lazo)
        frmruleblock1.Combo6(3).AddItem "term" + Str(lazo)
    Next lazo
End If

Case "bloquedereglas"
    rb_index = rb_index + 1
    If rb_index > 8 Then
        rb_index = 8
        MsgBox "Se excedió número máximo de bloques de reglas", vbExclamation, "Atención"
        GoTo Ignorebloque
    End If
    For lazo = 7 To rb_index
        Picture1(lazo).Left = 3392
        Picture1(lazo).Top = (lazo - 7) * Picture1(7).Height + (lazo - 6) * (Frame1.Height - (rb_index - 6) *
(Picture1(7).Height)) / (rb_index - 5)
    Next lazo
    If rb_index = 8 Then
        For lazo = 1 To input_index
            If Line1(lazo).Visible = True Then
                Line1(lazo).X1 = Picture1(lazo).Left + Picture1(lazo).Width
            End If
        Next lazo
    End If

```

```

        Line1(lazo).X2 = Picture1(7).Left
        Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)
        Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
    End If
Next lazo
For lazo = 10 To output_index
    If Line1(lazo).Visible = True Then
        Line1(lazo).X1 = Picture1(lazo).Left
        Line1(lazo).X2 = Picture1(7).Left + Picture1(7).Width
        Line1(lazo).Y1 = Picture1(lazo).Top + Round((Picture1(lazo).Height / 2), 0)
        Line1(lazo).Y2 = Picture1(7).Top + Round((Picture1(7).Height / 2), 0)
    End If
Next lazo
End If
Picture1(rb_index).Picture = LoadPicture("C:\Archivos de programa\Microsoft Visual
Studio\VB98\ruleblock.bmp")

Case "compilar"
    Call compile

Ignorebloque:
Ignoresalida:
Ignoreentrada:
End Select

End Sub

Private Sub compile()
    Dim RetVal
    Dim OutputFile As String
    dlgbase.CancelError = True
    On Error GoTo ManejoError:
    dlgbase.Flags = cd10FNHideReadOnly
    dlgbase.Filter = "Todos los archivos (*.*)|*.*|archivos ASM (*.asm)|*.asm"
    dlgbase.FilterIndex = 2
    dlgbase.ShowOpen
    OutputFile = "asm12 " & dlgbase.FileTitle & " -l -lu -hcs"
    RetVal = Shell(OutputFile, 1)

```

```

Exit Sub
ManejoError:
    MsgBox "Se eligió cancelar"
End Sub
Private Sub tools_compile_Click(Index As Integer)
Call compile
End Sub

```

```

Dim lazo, temporal As Integer
Dim Y1, Y2, Y3, Y4, y5 As Byte
Dim X1, X2, X3, X4, X5 As Byte
Dim X6, X7, X8, X9, X10 As Byte
Dim X11, X12, X13, X14, X15 As Byte
Dim X16, X17, X18, X19, X20 As Byte

```

```

Private Sub Command1_Click()
monitor1.Hide
End Sub

```

```

Private Sub Form_Activate()
For lazo = 1 To input_index
    Slider1(lazo).Top = Slider1(lazo).Height * (lazo - 1) + lazo * (3390 - input_index *
(Slider1(lazo).Height)) / (input_index + 1)
    'Picture1(lazo).Top = (lazo - 1) * Picture1(1).Height + lazo * (Frame1.Height - input_index *
(Picture1(1).Height)) / (input_index + 1)
    Slider1(lazo).Visible = True
Next lazo
End Sub

```

```

Private Sub Slider1_Scroll(Index As Integer)
'evaluación de los términos de la variable 1
If Index = 1 Then
X1 = Punto_pendiente(1, 1)
X2 = CByte((Val(frmtermino1prop.Text2.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X3 = CByte((Val(frmtermino1prop.Text3.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X4 = Punto_pendiente(1, 2)

```

```

'termino 1
If Slider1(Index).Value >= X1 And Slider1(Index).Value < X4 Then
    If Slider1(Index).Value >= X1 And Slider1(Index).Value < X2 Then
        Y1 = Punto_pendiente(1, 3) * (Slider1(Index).Value - Punto_pendiente(1, 1))
    End If
    If X3 = Punto_pendiente(1, 2) Then
        If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
            Y1 = Punto_pendiente(1, 4) * (Punto_pendiente(1, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X2 And Slider1(Index).Value < X3 Then
            Y1 = 255
        End If
    End If
    If Slider1(Index).Value >= X3 And Slider1(Index).Value <= X4 Then
        Y1 = Punto_pendiente(1, 4) * (Punto_pendiente(1, 2) - Slider1(Index).Value)
    End If
Else
    Y1 = 0
End If
If Y1 > 255 Then Y1 = 255
entrada1(0) = Y1
frmruleblock1.MSFlexGrid2.TextMatrix(0, 0) = Y1
' fin de termino 1

' termino 2
X1 = Punto_pendiente(6, 1)
X2 = CByte((Val(frmtermino2prop.Text2.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X3 = CByte((Val(frmtermino2prop.Text3.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X4 = Punto_pendiente(6, 2)

If Slider1(Index).Value >= X1 And Slider1(Index).Value <= X4 Then
    If Slider1(Index).Value >= X1 And Slider1(Index).Value < X2 Then
        Y1 = Punto_pendiente(6, 3) * (Slider1(Index).Value - Punto_pendiente(6, 1))
    End If
    If X3 = Punto_pendiente(6, 2) Then

```

```

    If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
        Y1 = Punto_pendiente(6, 4) * (Punto_pendiente(6, 2) - Slider1(Index).Value)
    End If
Else
    If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
        Y1 = 255
    End If
End If
If Slider1(Index).Value >= X3 And Slider1(Index).Value <= X4 Then
    Y1 = Punto_pendiente(6, 4) * (Punto_pendiente(6, 2) - Slider1(Index).Value)
End If
Else
    Y1 = 0
End If
If Y1 > 255 Then Y1 = 255
entrada1(1) = Y1
frmruleblock1.MSFlexGrid2.TextMatrix(1, 0) = Y1
'fin de termino 2

'termino 3
X1 = Punto_pendiente(11, 1)
X2 = CByte((Val(frmtermino3prop.Text2.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X3 = CByte((Val(frmtermino3prop.Text3.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X4 = Punto_pendiente(11, 2)
If Slider1(Index).Value >= Punto_pendiente(11, 1) And Slider1(Index).Value <= Punto_pendiente(11, 2)
Then
    If Slider1(Index).Value >= X1 And Slider1(Index).Value < X2 Then
        Y1 = Punto_pendiente(11, 3) * (Slider1(Index).Value - Punto_pendiente(11, 1))
    End If
    If X3 = Punto_pendiente(11, 2) And Punto_pendiente(11, 4) <> 0 Then
        If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
            Y1 = Punto_pendiente(11, 4) * (Punto_pendiente(11, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
            Y1 = 255
        End If
    End If

```

```

End If
If Slider1(Index).Value >= X3 And Slider1(Index).Value <= X4 And Punto_pendiente(11, 4) <> 0 Then
    Y1 = Punto_pendiente(11, 4) * (Punto_pendiente(11, 2) - Slider1(Index).Value)
End If
Else
    Y1 = 0
End If
If Y1 > 255 Then Y1 = 255
entrada1(2) = Y1
frmruleblock1.MSFlexGrid2.TextMatrix(2, 0) = Y1
'fin del termino 3

'termino4
X1 = Punto_pendiente(16, 1)
X2 = CByte((Val(frmtermino4prop.Text2.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X3 = CByte((Val(frmtermino4prop.Text3.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X4 = Punto_pendiente(16, 2)
If Slider1(Index).Value >= Punto_pendiente(16, 1) And Slider1(Index).Value <= Punto_pendiente(16, 2)
Then
    If Slider1(Index).Value >= X1 And Slider1(Index).Value < X2 Then
        Y1 = Punto_pendiente(16, 3) * (Slider1(Index).Value - Punto_pendiente(16, 1))
    End If
    If X3 = Punto_pendiente(16, 2) And Punto_pendiente(16, 4) <> 0 Then
        If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
            Y1 = Punto_pendiente(16, 4) * (Punto_pendiente(16, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
            Y1 = 255
        End If
    End If
    If Slider1(Index).Value >= X3 And Slider1(Index).Value <= X4 And Punto_pendiente(16, 4) <> 0 Then
        Y1 = Punto_pendiente(16, 4) * (Punto_pendiente(16, 2) - Slider1(Index).Value)
    End If
Else
    Y1 = 0
End If

```

```

If Y1 > 255 Then Y1 = 255
entrada1(3) = Y1
frmruleblock1.MSFlexGrid2.TextMatrix(3, 0) = Y1
'fin de termino 4

'termino 5
X1 = Punto_pendiente(21, 1)
X2 = CByte((Val(frmtermino5prop.Text2.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X3 = CByte((Val(frmtermino5prop.Text3.Text) / (Universo_entrada(1, 2) - Universo_entrada(1, 1))) * 255)
X4 = Punto_pendiente(21, 2)
If Slider1(Index).Value >= X1 And Slider1(Index).Value <= X4 Then
    If Slider1(Index).Value >= X1 And Slider1(Index).Value < X2 Then
        Y1 = Punto_pendiente(21, 3) * (Slider1(Index).Value - X1)
    End If
    If X3 = X4 And Punto_pendiente(21, 4) <> 0 Then
        If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
            Y1 = Punto_pendiente(21, 4) * (X4 - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X2 And Slider1(Index).Value <= X3 Then
            Y1 = 255
        End If
    End If
    If Slider1(Index).Value >= X3 And Slider1(Index).Value <= X4 And Punto_pendiente(21, 4) <> 0 Then
        Y1 = Punto_pendiente(21, 4) * (X4 - Slider1(Index).Value)
    End If
Else
    Y1 = 0
End If
If Y1 > 255 Then Y1 = 255
entrada1(4) = Y1
frmruleblock1.MSFlexGrid2.TextMatrix(4, 0) = Y1
End If

'evaluación de los términos de la variable 2
If Index = 2 Then
'variables para el termino 1

```

```

X5 = Punto_pendiente(2, 1)
X6 = CByte((Val(frmtermino1prop2.Text2.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
X7 = CByte((Val(frmtermino1prop2.Text3.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
X4 = Punto_pendiente(2, 2)

'termino 1
If Slider1(Index).Value >= X5 And Slider1(Index).Value < X4 Then
    If Slider1(Index).Value >= X5 And Slider1(Index).Value < X6 Then
        Y2 = Punto_pendiente(2, 3) * (Slider1(Index).Value - Punto_pendiente(2, 1))
    End If
    If X7 = Punto_pendiente(2, 2) Then
        If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
            Y2 = Punto_pendiente(2, 4) * (Punto_pendiente(2, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X6 And Slider1(Index).Value < X7 Then
            Y2 = 255
        End If
    End If
    If Slider1(Index).Value >= X7 And Slider1(Index).Value <= X4 Then
        Y2 = Punto_pendiente(2, 4) * (Punto_pendiente(2, 2) - Slider1(Index).Value)
    End If
Else
    Y2 = 0
End If
If Y2 > 255 Then Y2 = 255
entrada2(0) = Y2
frmruleblock1.MSFlexGrid2.TextMatrix(0, 1) = Y2
' fin de termino 1

'termino 2
X5 = Punto_pendiente(7, 1)
X6 = CByte((Val(frmtermino2prop2.Text2.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)

```



```
X7 = CByte((Val(frmtermino2prop2.Text3.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
```

```
X8 = Punto_pendiente(7, 2)
```

```
If Slider1(Index).Value >= X5 And Slider1(Index).Value <= X8 Then
```

```
    If Slider1(Index).Value >= X5 And Slider1(Index).Value < X6 Then
```

```
        Y2 = Punto_pendiente(7, 3) * (Slider1(Index).Value - Punto_pendiente(7, 1))
```

```
    End If
```

```
    If X7 = Punto_pendiente(7, 2) Then
```

```
        If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
```

```
            Y2 = Punto_pendiente(7, 4) * (Punto_pendiente(7, 2) - Slider1(Index).Value)
```

```
        End If
```

```
    Else
```

```
        If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
```

```
            Y2 = 255
```

```
        End If
```

```
    End If
```

```
If Slider1(Index).Value >= X7 And Slider1(Index).Value <= X8 Then
```

```
    Y2 = Punto_pendiente(7, 4) * (Punto_pendiente(7, 2) - Slider1(Index).Value)
```

```
End If
```

```
Else
```

```
    Y2 = 0
```

```
End If
```

```
If Y2 > 255 Then Y2 = 255
```

```
entrada2(1) = Y2
```

```
frmruleblock1.MSFlexGrid2.TextMatrix(1, 1) = Y2
```

```
'fin de termino 2
```

```
'termino 3
```

```
X5 = Punto_pendiente(12, 1)
```

```
X6 = CByte((Val(frmtermino3prop2.Text2.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
```

```
X7 = CByte((Val(frmtermino3prop2.Text3.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
```

```
X8 = Punto_pendiente(12, 2)
```

```
If Slider1(Index).Value >= Punto_pendiente(12, 1) And Slider1(Index).Value <= Punto_pendiente(12, 2)
```

```
Then
```

```

If Slider1(Index).Value >= X5 And Slider1(Index).Value < X6 Then
    Y2 = Punto_pendiente(12, 3) * (Slider1(Index).Value - Punto_pendiente(12, 1))
End If
If X7 = Punto_pendiente(12, 2) And Punto_pendiente(12, 4) <> 0 Then
    If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
        Y2 = Punto_pendiente(12, 4) * (Punto_pendiente(12, 2) - Slider1(Index).Value)
    End If
Else
    If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
        Y2 = 255
    End If
End If
If Slider1(Index).Value >= X7 And Slider1(Index).Value <= X8 And Punto_pendiente(12, 4) <> 0 Then
    Y2 = Punto_pendiente(12, 4) * (Punto_pendiente(12, 2) - Slider1(Index).Value)
End If
Else
    Y2 = 0
End If
If Y2 > 255 Then Y2 = 255
entrada2(2) = Y2
frmruleblock1.MSFlexGrid2.TextMatrix(2, 1) = Y2
'fin del termino 3

'termino4
X5 = Punto_pendiente(17, 1)
X6 = CByte((Val(frmtermino4prop2.Text2.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
X7 = CByte((Val(frmtermino4prop2.Text3.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
X8 = Punto_pendiente(17, 2)
If Slider1(Index).Value >= Punto_pendiente(17, 1) And Slider1(Index).Value <= Punto_pendiente(17, 2)
Then
    If Slider1(Index).Value >= X5 And Slider1(Index).Value < X6 Then
        Y2 = Punto_pendiente(17, 3) * (Slider1(Index).Value - Punto_pendiente(17, 1))
    End If
    If X7 = Punto_pendiente(17, 2) And Punto_pendiente(17, 4) <> 0 Then
        If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then

```

```

        Y2 = Punto_pendiente(17, 4) * (Punto_pendiente(17, 2) - Slider1(Index).Value)
    End If
Else
    If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
        Y2 = 255
    End If
End If
If Slider1(Index).Value >= X7 And Slider1(Index).Value <= X8 And Punto_pendiente(17, 4) <> 0 Then
    Y2 = Punto_pendiente(17, 4) * (Punto_pendiente(17, 2) - Slider1(Index).Value)
End If
Else
    Y2 = 0
End If
If Y2 > 255 Then Y2 = 255
entrada2(3) = Y2
frmruleblock1.MSFlexGrid2.TextMatrix(3, 1) = Y2
'fin de termino 4

'termino 5
X5 = Punto_pendiente(22, 1)
X6 = CByte((Val(frmtermino5prop2.Text2.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
X7 = CByte((Val(frmtermino5prop2.Text3.Text) / (Universo_entrada(2, 2) - Universo_entrada(2, 1))) *
255)
X8 = Punto_pendiente(22, 2)
If Slider1(Index).Value >= X5 And Slider1(Index).Value <= X8 Then
    If Slider1(Index).Value >= X5 And Slider1(Index).Value < X6 Then
        Y2 = Punto_pendiente(22, 3) * (Slider1(Index).Value - X5)
    End If
    If X7 = X8 And Punto_pendiente(22, 4) <> 0 Then
        If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
            Y2 = Punto_pendiente(22, 4) * (X8 - Slider1(Index).Value)
        End If
    End If
Else
    If Slider1(Index).Value >= X6 And Slider1(Index).Value <= X7 Then
        Y2 = 255
    End If

```

```

End If
If Slider1(Index).Value >= X7 And Slider1(Index).Value <= X8 And Punto_pendiente(22, 4) <> 0 Then
    Y2 = Punto_pendiente(22, 4) * (X8 - Slider1(Index).Value)
End If
Else
    Y2 = 0
End If
If Y2 > 255 Then Y2 = 255
entrada2(4) = Y2
frmruleblock1.MSFlexGrid2.TextMatrix(4, 1) = Y2
End If

'evaluación de los términos de la variable 3
If Index = 3 Then
'variables para el termino 1
X9 = Punto_pendiente(3, 1)
X10 = CByte((Val(frmtermino1prop3.Text2.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X11 = CByte((Val(frmtermino1prop3.Text3.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X12 = Punto_pendiente(3, 2)

'termino 1
If Slider1(Index).Value >= X9 And Slider1(Index).Value < X12 Then
    If Slider1(Index).Value >= X9 And Slider1(Index).Value < X10 Then
        Y3 = Punto_pendiente(3, 3) * (Slider1(Index).Value - Punto_pendiente(3, 1))
    End If
    If X11 = Punto_pendiente(3, 2) Then
        If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then
            Y3 = Punto_pendiente(3, 4) * (Punto_pendiente(3, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X10 And Slider1(Index).Value < X11 Then
            Y3 = 255
        End If
    End If
    If Slider1(Index).Value >= X11 And Slider1(Index).Value <= X12 Then

```

```

        Y3 = Punto_pendiente(3, 4) * (Punto_pendiente(3, 2) - Slider1(Index).Value)
    End If
Else
    Y3 = 0
End If
If Y3 > 255 Then Y3 = 255
entrada3(0) = Y3
' fin de termino 1

' termino 2
X9 = Punto_pendiente(8, 1)
X10 = CByte((Val(frmtermino2prop3.Text2.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X11 = CByte((Val(frmtermino2prop3.Text3.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X12 = Punto_pendiente(8, 2)

If Slider1(Index).Value >= X9 And Slider1(Index).Value <= X12 Then
    If Slider1(Index).Value >= X9 And Slider1(Index).Value < X10 Then
        Y3 = Punto_pendiente(8, 3) * (Slider1(Index).Value - Punto_pendiente(8, 1))
    End If
    If X11 = Punto_pendiente(8, 2) Then
        If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then
            Y3 = Punto_pendiente(8, 4) * (Punto_pendiente(8, 2) - Slider1(Index).Value)
        End If
    End If
Else
    If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then
        Y3 = 255
    End If
End If
If Slider1(Index).Value >= X11 And Slider1(Index).Value <= X12 Then
    Y3 = Punto_pendiente(8, 4) * (Punto_pendiente(8, 2) - Slider1(Index).Value)
End If
Else
    Y3 = 0
End If
If Y3 > 255 Then Y3 = 255

```

entrada3(1) = Y3

'fin de termino 2

'termino 3

X9 = Punto_pendiente(13, 1)

X10 = CByte((Val(frmtermino3prop3.Text2.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) * 255)

X11 = CByte((Val(frmtermino3prop3.Text3.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) * 255)

X12 = Punto_pendiente(13, 2)

If Slider1(Index).Value >= Punto_pendiente(13, 1) And Slider1(Index).Value <= Punto_pendiente(13, 2)

Then

If Slider1(Index).Value >= X9 And Slider1(Index).Value < X10 Then

Y3 = Punto_pendiente(13, 3) * (Slider1(Index).Value - Punto_pendiente(13, 1))

End If

If X11 = Punto_pendiente(13, 2) And Punto_pendiente(13, 4) <> 0 Then

If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then

Y3 = Punto_pendiente(13, 4) * (Punto_pendiente(13, 2) - Slider1(Index).Value)

End If

Else

If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then

Y3 = 255

End If

End If

If Slider1(Index).Value >= X11 And Slider1(Index).Value <= X12 And Punto_pendiente(13, 4) <> 0

Then

Y3 = Punto_pendiente(13, 4) * (Punto_pendiente(13, 2) - Slider1(Index).Value)

End If

Else

Y3 = 0

End If

If Y3 > 255 Then Y3 = 255

entrada3(2) = Y3

'fin del termino 3

'termino4

X9 = Punto_pendiente(18, 1)

```

X10 = CByte((Val(frmtermino4prop3.Text2.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X11 = CByte((Val(frmtermino4prop3.Text3.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X12 = Punto_pendiente(18, 2)
If Slider1(Index).Value >= Punto_pendiente(18, 1) And Slider1(Index).Value <= Punto_pendiente(18, 2)
Then
    If Slider1(Index).Value >= X9 And Slider1(Index).Value < X10 Then
        Y3 = Punto_pendiente(18, 3) * (Slider1(Index).Value - Punto_pendiente(18, 1))
    End If
    If X11 = Punto_pendiente(18, 2) And Punto_pendiente(18, 4) <> 0 Then
        If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then
            Y3 = Punto_pendiente(18, 4) * (Punto_pendiente(18, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then
            Y3 = 255
        End If
    End If
    If Slider1(Index).Value >= X11 And Slider1(Index).Value <= X12 And Punto_pendiente(18, 4) <> 0
Then
        Y3 = Punto_pendiente(18, 4) * (Punto_pendiente(18, 2) - Slider1(Index).Value)
    End If
Else
    Y3 = 0
End If
If Y3 > 255 Then Y3 = 255
entrada3(3) = Y3
'fin de termino 4

'termino 5
X9 = Punto_pendiente(23, 1)
X10 = CByte((Val(frmtermino5prop3.Text2.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X11 = CByte((Val(frmtermino5prop3.Text3.Text) / (Universo_entrada(3, 2) - Universo_entrada(3, 1))) *
255)
X12 = Punto_pendiente(23, 2)

```

```

If Slider1(Index).Value >= X9 And Slider1(Index).Value <= X12 Then
  If Slider1(Index).Value >= X9 And Slider1(Index).Value < X10 Then
    Y3 = Punto_pendiente(23, 3) * (Slider1(Index).Value - X9)
  End If
  If X11 = X12 And Punto_pendiente(23, 4) <> 0 Then
    If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then
      Y3 = Punto_pendiente(23, 4) * (X12 - Slider1(Index).Value)
    End If
  Else
    If Slider1(Index).Value >= X10 And Slider1(Index).Value <= X11 Then
      Y3 = 255
    End If
  End If
  If Slider1(Index).Value >= X11 And Slider1(Index).Value <= X12 And Punto_pendiente(23, 4) <> 0
Then
    Y3 = Punto_pendiente(23, 4) * (X12 - Slider1(Index).Value)
  End If
Else
  Y3 = 0
End If
If Y3 > 255 Then Y3 = 255
entrada3(4) = Y3
End If

```

'evaluación de los términos de la variable 4

```

If Index = 4 Then

```

'variables para el termino 1

```

X13 = Punto_pendiente(4, 1)

```

```

X14 = CByte((Val(frmtermino1prop4.Text2.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)

```

```

X15 = CByte((Val(frmtermino1prop4.Text3.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)

```

```

X16 = Punto_pendiente(4, 2)

```

'termino 1

```

If Slider1(Index).Value >= X13 And Slider1(Index).Value < X16 Then

```

```

  If Slider1(Index).Value >= X13 And Slider1(Index).Value < X14 Then

```



```

    Y4 = Punto_pendiente(4, 3) * (Slider1(Index).Value - Punto_pendiente(4, 1))
End If
If X15 = Punto_pendiente(4, 2) Then
    If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
        Y4 = Punto_pendiente(4, 4) * (Punto_pendiente(4, 2) - Slider1(Index).Value)
    End If
Else
    If Slider1(Index).Value >= X14 And Slider1(Index).Value < X15 Then
        Y4 = 255
    End If
End If
If Slider1(Index).Value >= X15 And Slider1(Index).Value <= X16 Then
    Y4 = Punto_pendiente(4, 4) * (Punto_pendiente(4, 2) - Slider1(Index).Value)
End If
Else
    Y4 = 0
End If
If Y4 > 255 Then Y4 = 255
entrada4(0) = Y4
' fin de termino 1

' termino 2
X13 = Punto_pendiente(9, 1)
X14 = CByte((Val(frmtermino2prop4.Text2.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X15 = CByte((Val(frmtermino2prop4.Text3.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X16 = Punto_pendiente(9, 2)

If Slider1(Index).Value >= X13 And Slider1(Index).Value <= X16 Then
    If Slider1(Index).Value >= X13 And Slider1(Index).Value < X14 Then
        Y4 = Punto_pendiente(9, 3) * (Slider1(Index).Value - Punto_pendiente(9, 1))
    End If
    If X15 = Punto_pendiente(9, 2) Then
        If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
            Y4 = Punto_pendiente(9, 4) * (Punto_pendiente(9, 2) - Slider1(Index).Value)
        End If
    End If

```

```

Else
    If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
        Y4 = 255
    End If
End If
If Slider1(Index).Value >= X15 And Slider1(Index).Value <= X16 Then
    Y4 = Punto_pendiente(9, 4) * (Punto_pendiente(9, 2) - Slider1(Index).Value)
End If
Else
    Y4 = 0
End If
If Y4 > 255 Then Y4 = 255
entrada4(1) = Y4
'fin de termino 2

'termino 3
X13 = Punto_pendiente(14, 1)
X14 = CByte((Val(frmtermino3prop4.Text2.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X15 = CByte((Val(frmtermino3prop4.Text3.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X16 = Punto_pendiente(14, 2)
If Slider1(Index).Value >= Punto_pendiente(14, 1) And Slider1(Index).Value <= Punto_pendiente(14, 2)
Then
    If Slider1(Index).Value >= X13 And Slider1(Index).Value < X14 Then
        Y4 = Punto_pendiente(14, 3) * (Slider1(Index).Value - Punto_pendiente(14, 1))
    End If
    If X15 = Punto_pendiente(14, 2) And Punto_pendiente(14, 4) <> 0 Then
        If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
            Y4 = Punto_pendiente(14, 4) * (Punto_pendiente(14, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
            Y4 = 255
        End If
    End If
End If

```

```

    If Slider1(Index).Value >= X15 And Slider1(Index).Value <= X16 And Punto_pendiente(14, 4) <> 0
Then
    Y4 = Punto_pendiente(14, 4) * (Punto_pendiente(14, 2) - Slider1(Index).Value)
End If
Else
    Y4 = 0
End If
If Y4 > 255 Then Y4 = 255
entrada4(2) = Y4
'fin del termino 3

'termino4
X13 = Punto_pendiente(19, 1)
X14 = CByte((Val(frmtermino4prop4.Text2.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X15 = CByte((Val(frmtermino4prop4.Text3.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X16 = Punto_pendiente(19, 2)
If Slider1(Index).Value >= Punto_pendiente(19, 1) And Slider1(Index).Value <= Punto_pendiente(19, 2)
Then
    If Slider1(Index).Value >= X13 And Slider1(Index).Value < X14 Then
        Y4 = Punto_pendiente(19, 3) * (Slider1(Index).Value - Punto_pendiente(19, 1))
    End If
    If X15 = Punto_pendiente(19, 2) And Punto_pendiente(19, 4) <> 0 Then
        If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
            Y4 = Punto_pendiente(19, 4) * (Punto_pendiente(19, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
            Y4 = 255
        End If
    End If
    If Slider1(Index).Value >= X15 And Slider1(Index).Value <= X16 And Punto_pendiente(19, 4) <> 0
Then
    Y4 = Punto_pendiente(19, 4) * (Punto_pendiente(19, 2) - Slider1(Index).Value)
End If
Else

```

```

    Y4 = 0
End If
If Y4 > 255 Then Y4 = 255
entrada4(3) = Y4
'fin de termino 4

'termino 5
X13 = Punto_pendiente(24, 1)
X14 = CByte((Val(frmtermino5prop4.Text2.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X15 = CByte((Val(frmtermino5prop4.Text3.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X16 = Punto_pendiente(24, 2)
If Slider1(Index).Value >= X13 And Slider1(Index).Value <= X16 Then
    If Slider1(Index).Value >= X13 And Slider1(Index).Value < X14 Then
        Y4 = Punto_pendiente(24, 3) * (Slider1(Index).Value - X13)
    End If
    If X15 = X16 And Punto_pendiente(24, 4) <> 0 Then
        If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
            Y4 = Punto_pendiente(24, 4) * (X16 - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X14 And Slider1(Index).Value <= X15 Then
            Y4 = 255
        End If
    End If
    If Slider1(Index).Value >= X15 And Slider1(Index).Value <= X16 And Punto_pendiente(24, 4) <> 0
Then
        Y4 = Punto_pendiente(24, 4) * (X16 - Slider1(Index).Value)
    End If
Else
    Y4 = 0
End If
If Y4 > 255 Then Y4 = 255
entrada4(4) = Y4
End If

```

'evaluación de los términos de la variable 5

If Index = 5 Then

'variables para el termino 1

X17 = Punto_pendiente(5, 1)

X18 = CByte((Val(frmtermino1prop5.Text2.Text) / (Universo_entrada(5, 2) - Universo_entrada(5, 1))) * 255)

X19 = CByte((Val(frmtermino1prop5.Text3.Text) / (Universo_entrada(5, 2) - Universo_entrada(5, 1))) * 255)

X20 = Punto_pendiente(5, 2)

'termino 1

If Slider1(Index).Value >= X17 And Slider1(Index).Value < X20 Then

If Slider1(Index).Value >= X17 And Slider1(Index).Value < X18 Then

y5 = Punto_pendiente(5, 3) * (Slider1(Index).Value - Punto_pendiente(5, 1))

End If

If X19 = Punto_pendiente(5, 2) Then

If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then

y5 = Punto_pendiente(5, 4) * (Punto_pendiente(5, 2) - Slider1(Index).Value)

End If

Else

If Slider1(Index).Value >= X18 And Slider1(Index).Value < X19 Then

y5 = 255

End If

End If

If Slider1(Index).Value >= X19 And Slider1(Index).Value <= X20 Then

y5 = Punto_pendiente(5, 4) * (Punto_pendiente(5, 2) - Slider1(Index).Value)

End If

Else

y5 = 0

End If

If y5 > 255 Then y5 = 255

entrada5(0) = y5

' fin de termino 1

' termino 2

X17 = Punto_pendiente(10, 1)

```
X18 = CByte((Val(frmtermino2prop5.Text2.Text) / (Universo_entrada(5, 2) - Universo_entrada(5, 1))) * 255)
```

```
X19 = CByte((Val(frmtermino2prop5.Text3.Text) / (Universo_entrada(5, 2) - Universo_entrada(5, 1))) * 255)
```

```
X20 = Punto_pendiente(10, 2)
```

```
If Slider1(Index).Value >= X17 And Slider1(Index).Value <= X20 Then
```

```
  If Slider1(Index).Value >= X17 And Slider1(Index).Value < X18 Then
```

```
    y5 = Punto_pendiente(10, 3) * (Slider1(Index).Value - Punto_pendiente(10, 1))
```

```
  End If
```

```
  If X19 = Punto_pendiente(10, 2) Then
```

```
    If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then
```

```
      y5 = Punto_pendiente(10, 4) * (Punto_pendiente(10, 2) - Slider1(Index).Value)
```

```
    End If
```

```
  Else
```

```
    If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then
```

```
      y5 = 255
```

```
    End If
```

```
  End If
```

```
  If Slider1(Index).Value >= X19 And Slider1(Index).Value <= X20 Then
```

```
    y5 = Punto_pendiente(10, 4) * (Punto_pendiente(10, 2) - Slider1(Index).Value)
```

```
  End If
```

```
Else
```

```
  y5 = 0
```

```
End If
```

```
If y5 > 255 Then y5 = 255
```

```
entrada5(1) = y5
```

```
'fin de termino 2
```

```
'termino 3
```

```
X17 = Punto_pendiente(15, 1)
```

```
X18 = CByte((Val(frmtermino3prop5.Text2.Text) / (Universo_entrada(5, 2) - Universo_entrada(5, 1))) * 255)
```

```
X19 = CByte((Val(frmtermino3prop5.Text3.Text) / (Universo_entrada(4, 2) - Universo_entrada(5, 1))) * 255)
```

```
X20 = Punto_pendiente(15, 2)
```

```

If Slider1(Index).Value >= Punto_pendiente(15, 1) And Slider1(Index).Value <= Punto_pendiente(15, 2)
Then
    If Slider1(Index).Value >= X17 And Slider1(Index).Value < X18 Then
        y5 = Punto_pendiente(15, 3) * (Slider1(Index).Value - Punto_pendiente(15, 1))
    End If
    If X19 = Punto_pendiente(15, 2) And Punto_pendiente(15, 4) <> 0 Then
        If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then
            y5 = Punto_pendiente(15, 4) * (Punto_pendiente(15, 2) - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then
            y5 = 255
        End If
    End If
    If Slider1(Index).Value >= X19 And Slider1(Index).Value <= X20 And Punto_pendiente(15, 4) <> 0
Then
        y5 = Punto_pendiente(15, 4) * (Punto_pendiente(15, 2) - Slider1(Index).Value)
    End If
Else
    y5 = 0
End If
If y5 > 255 Then y5 = 255
entrada5(2) = y5
'fin del termino 3

```

'termino4

```

X17 = Punto_pendiente(19, 1)
X18 = CByte((Val(frmtermino4prop5.Text2.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X19 = CByte((Val(frmtermino4prop5.Text3.Text) / (Universo_entrada(4, 2) - Universo_entrada(4, 1))) *
255)
X20 = Punto_pendiente(19, 2)
If Slider1(Index).Value >= Punto_pendiente(19, 1) And Slider1(Index).Value <= Punto_pendiente(19, 2)
Then
    If Slider1(Index).Value >= X17 And Slider1(Index).Value < X18 Then
        y5 = Punto_pendiente(19, 3) * (Slider1(Index).Value - Punto_pendiente(19, 1))
    End If

```

```

If X19 = Punto_pendiente(19, 2) And Punto_pendiente(19, 4) <> 0 Then
    If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then
        y5 = Punto_pendiente(19, 4) * (Punto_pendiente(19, 2) - Slider1(Index).Value)
    End If
Else
    If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then
        y5 = 255
    End If
End If
If Slider1(Index).Value >= X19 And Slider1(Index).Value <= X20 And Punto_pendiente(19, 4) <> 0
Then
    y5 = Punto_pendiente(19, 4) * (Punto_pendiente(19, 2) - Slider1(Index).Value)
End If
Else
    y5 = 0
End If
If y5 > 255 Then y5 = 255
entrada5(3) = y5
'fin de termino 4

'termino 5
X17 = Punto_pendiente(25, 1)
X18 = CByte((Val(frmtermino5prop5.Text2.Text) / (Universo_entrada(5, 2) - Universo_entrada(5, 1))) *
255)
X19 = CByte((Val(frmtermino5prop5.Text3.Text) / (Universo_entrada(5, 2) - Universo_entrada(5, 1))) *
255)
X20 = Punto_pendiente(25, 2)
If Slider1(Index).Value >= X17 And Slider1(Index).Value <= X20 Then
    If Slider1(Index).Value >= X17 And Slider1(Index).Value < X18 Then
        y5 = Punto_pendiente(25, 3) * (Slider1(Index).Value - X17)
    End If
    If X19 = X20 And Punto_pendiente(25, 4) <> 0 Then
        If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then
            y5 = Punto_pendiente(25, 4) * (X20 - Slider1(Index).Value)
        End If
    Else
        If Slider1(Index).Value >= X18 And Slider1(Index).Value <= X19 Then

```



```

        y5 = 255
    End If
End If
If Slider1(Index).Value >= X19 And Slider1(Index).Value <= X20 And Punto_pendiente(25, 4) <> 0
Then
    y5 = Punto_pendiente(25, 4) * (X20 - Slider1(Index).Value)
End If
Else
    y5 = 0
End If
If y5 > 255 Then y5 = 255
entrada5(4) = y5
End If

For lazo = 0 To 4
    salida1(lazo) = 0
Next lazo
temporal = 1

If input_index = 1 Then
    For lazo = 1 To input_index
        temporal = temporal * matriz_indices(lazo, 2)
    Next lazo
    For lazo = 0 To temporal - 1
        salida1(matriz_puntero(lazo, 5)) = FuzzyOr(salida1(matriz_puntero(lazo, 5)),
        entrada1(matriz_puntero(lazo, 0)))
    Next lazo
    For lazo = 0 To 4
        frmruleblock1.MSFlexGrid2.TextMatrix(lazo, 2) = salida1(lazo)
    Next lazo
End If
If input_index >= 2 Then
    For lazo = 1 To input_index
        temporal = temporal * matriz_indices(lazo, 2)
    Next lazo
    For lazo = 0 To temporal - 1

```

```

        salida1(matriz_puntero(lazo, 5)) = FuzzyOr(salida1(matriz_puntero(lazo, 5)),
FuzzyAnd(entrada1(matriz_puntero(lazo, 0)), entrada2(matriz_puntero(lazo, 1))))

```

```

    Next lazo

```

```

    For lazo = 0 To 4

```

```

        frmruleblock1.MSFlexGrid2.TextMatrix(lazo, 2) = salida1(lazo)

```

```

    Next lazo

```

```

End If

```

```

End Sub

```

```

Public matriz_puntos(1 To 5, 1 To 5, 1 To 4, 1 To 4) As Integer

```

```

Public matriz_indices(1 To 5, 1 To 6) As Integer

```

```

Public VarNames(1 To 5) As String

```

```

Public OutNames(1 To 3) As String

```

```

Public Universo_entrada(1 To 5, 1 To 2) As Variant

```

```

Public Universo_salida(1 To 3, 1 To 2) As Variant

```

```

Public Punto_pendiente(1 To 25, 1 To 4) As Byte

```

```

Public input_index As Integer

```

```

Public rb_index As Integer

```

```

Public output_index As Integer

```

```

Public matriz_puntero(0 To 169, 0 To 7) As Integer

```

```

Public entrada1(0 To 4), entrada2(0 To 4), entrada3(0 To 4), entrada4(0 To 4), entrada5(0 To 4) As Byte

```

```

Public salida1(0 To 4) As Byte

```

```

Public Function FuzzyOr(ByVal operand1 As Byte, ByVal operand2 As Byte) As Byte

```

```

    If operand1 >= operand2 Then FuzzyOr = operand1

```

```

    If operand2 > operand1 Then FuzzyOr = operand2

```

```

End Function

```

```

Public Function FuzzyAnd(ByVal operand1 As Byte, ByVal operand2 As Byte) As Byte

```

```

    If operand1 <= operand2 Then FuzzyAnd = operand1

```

```

    If operand2 < operand1 Then FuzzyAnd = operand2

```

```

End Function

```

A.2 MANUAL DEL USUARIO.

VISIONES DE LA LOGICA DIFUSA.

“ Es la marca de una mente instruida el descansar satisfecho con que el grado de precisión que la naturaleza de los sujetos admite es tal, y no buscar exactitud donde solo una aproximación de la verdad es posible”

Aristóteles, Antiguo Filósofo griego

“La Precisión no es verdadera”

Henri Matisse, Pintor Impresionista

“Tan lejos como las leyes de la matemática se refieren a la realidad, así no son certeras. Y así de lejos como están de la realidad, no se refieren a la realidad”

Albert Einstein, Físico y Matemático

“En casi cada caso se puede construir el mismo producto sin la lógica difusa, pero la lógica difusa es más rápida y barata”

Prof. Lofti Zadeh, UC Berkeley, Inventor de la Lógica Difusa

“La lógica difusa permite centrarse en resolver el problema en lugar de como resolver el problema”

Wayne Needham, Intel Corporation

“Hoy, un pre-requisito para el una solución rápida y competente de lógica difusa son las herramientas de software que tienen componentes de ser amigables con el usuario así como componentes científicos en sí

Prof. Zimmermann, Presidente de la Asociación Internacional de Lógica Difusa

“Si es cero es cero”

Tec. Br. Carlos Cabrera, Universidad Don Bosco

ANTES DE EMPEZAR

Este manual provee de una introducción al Simulador Difuso, incluyendo lo que se necesita para la instalación, así como para utilizar el programa. Esta guía esta diseñada para los programadores que deseen implementar sistemas de lógica difusa y su objetivo principal es brindar la ayuda necesaria para lograr que los sistemas simulados e implementados funcionen de una manera apropiada.

COMO USAR ESTE MANUAL

El manual de Usuario del Simulador difuso está compuesto por los siguientes capítulos:

Capítulo Uno: Provee una descripción breve de las características principales del Simulador Difuso.

Capítulo Dos: Describe la instalación del Software en una Computadora Personal PC así como el proceso llevado a cabo para la puesta en marcha de la aplicación.

Capítulo Tres: Explica los pasos para el desarrollo de un sistema de control difuso. Partiendo de la definición de entradas, la generación del bloque de reglas y la definición de salidas.

Capítulo Cuatro: Brinda una breve descripción de la Lógica Difusa y su teoría básica así como la aplicación de estas en el controlador difuso.

Capítulo Cinco: Presenta una descripción del soporte que para Lógica Difusa brinda el microcontrolador Motorola 68HC12.

CAPITULO 1. Introducción.

Este capítulo describe los componentes del Simulador Difuso y provee una breve introducción de los componentes del mismo.

Contenido

- Acerca del Simulador Difuso
- Entorno de programación
- El microcontrolador 68hc12

1.1 Acerca del Simulador Difuso

La operación de cualquier industria va mucho mas allá de la simple actividad de las personas, la transformación y la venta de materiales y equipos.

Es también la administración de los datos de la planta, ya que los procesos solo son óptimos cuando se conoce como deben operar los controladores así como las acciones que se deben de tomar ante distintas situaciones, ya sean estas acciones automáticas o acciones tomadas sobre la base de una decisión humana.

El simulador difuso es una herramienta industrial capaz de brindar la ayuda necesaria al instrumentista para desarrollar proyectos de lógica difusa basándose en el soporte que para ello brinda el microcontrolador Motorola 68HC12.

A través del simulador difuso se puede tener una estimación del funcionamiento del sistema antes de implementarlo y luego descargar el programa generado en una tarjeta 68HC12 para la puesta en marcha del controlador.

1.2 Entorno de programación

El entorno de programación para el simulador difuso está realizado en Visual Basic® pero no tiene relación alguna con la programación en VB ni con la empresa Microsoft®.

El desarrollo de la aplicación de desarrollo de controladores difusos ha sido diseñada con el propósito de facilitar el proceso de puesta en marcha de un controlador difuso.

Para el uso de la Aplicación del Simulador Difuso no es necesario el conocimiento de un lenguaje de programación.

1.3 El microcontrolador 68hc12

El microcontrolador Motorola 68hc12 es un microcontrolador de amplia difusión en los sistemas de control industrial utilizado por varias compañías para la implementación de sus funciones en equipos de distinta índole, es además un microcontrolador que ya posee intrínsecamente un apartado de soporte para lógica difusa por medio de instrucciones dedicadas para tal fin, las cuales minimizan tanto la programación como el tiempo de ejecución de los programas de control.

1.4 Requerimientos de Hardware y Software

Esta sección describe todos los requerimientos de hardware y software que deben cumplirse para un buen funcionamiento del simulador difuso. Asegúrese antes de iniciar la instalación del sistema que se cumplen todos y cada uno de los requerimientos.

Requerimientos de Hardware:

- CPU Intel Pentium I o mayor.
- 128 Mbytes de memoria RAM
- Puerto serial, o USB mas kit de conversión de USB a serial (opcional)
- Lector de discos compactos CD-ROM

Requerimientos de Software:

- Sistema Operativo Windows 98 Segunda Edición o superior

CAPITULO 2. Instalación del Simulador Difuso

El Simulador Difuso está disponible para su instalación a través de su Disco de Instalación contenido en un CD-ROM.

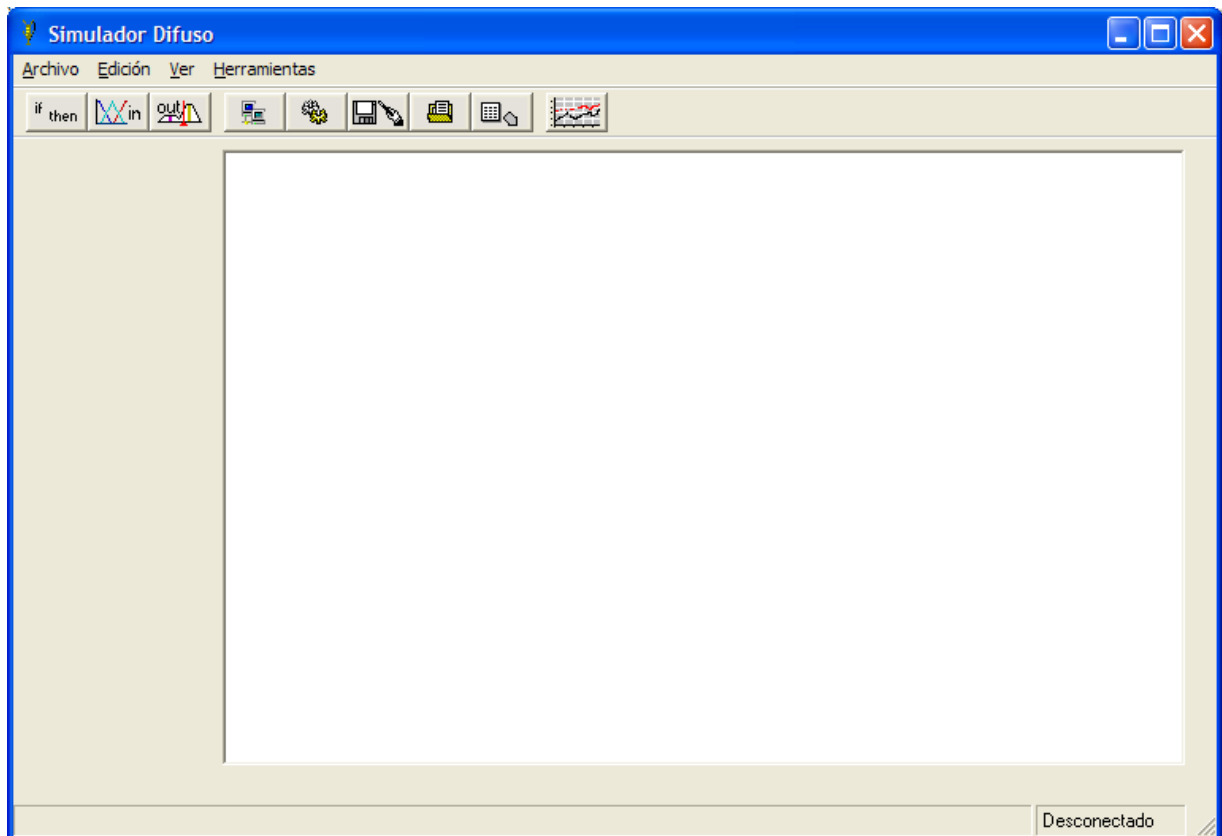
La instalación de la Aplicación es automática una vez se introduce el CD-ROM de instalación si está seleccionada la opción de Autorun para el lector de CDs. De no ser así, se debe acceder a Mi PC y ejecutar el programa Setup contenido en el CD-ROM.

CAPITULO 3. Uso del Simulador Difuso

Una vez instalada la aplicación se puede abrir dicha aplicación y ejecutarse de una manera autónoma.

Contenido:

- Pantalla Principal



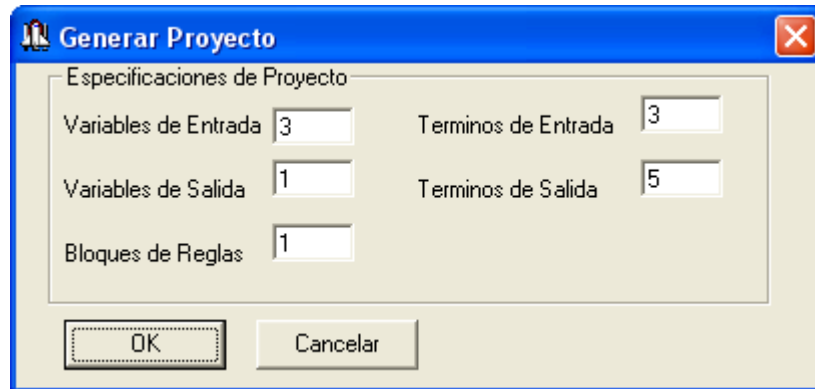
Pantalla 1

Los componentes de la pantalla 1 son los de la pantalla principal y son básicamente dos: la barra de menú y la pantalla de trabajo. En la barra de menú se encuentran las siguientes opciones: Archivo, Edición, Ver y Herramientas.

Dentro de la primera opción Archivo se encuentra un submenú del cual se puede elegir entre **Nuevo, Abrir, Cerrar, Guardar, Guardar Como y Salir**.

Cuando se elige **Nuevo** se puede configurar un control difuso definiendo los siguientes parámetros:

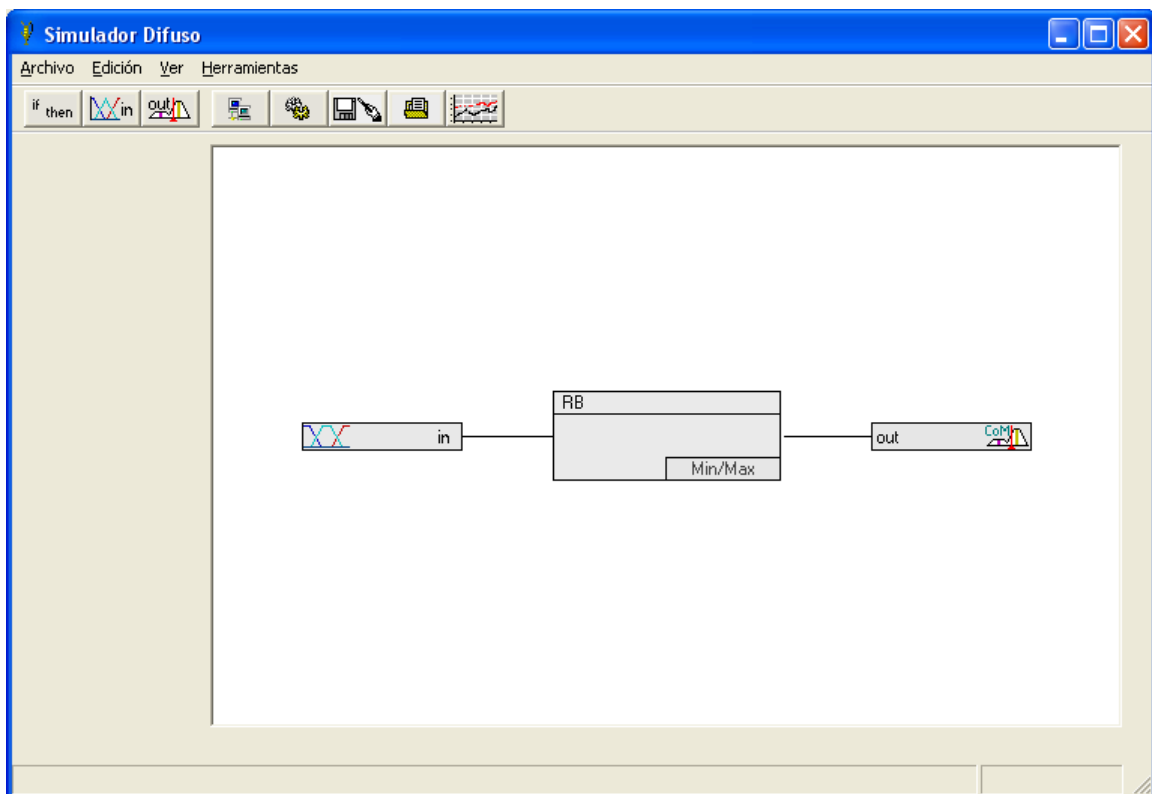
- Definición de las entradas
- Determinación de los bloques de reglas
- Definición de las salidas difusas
- Edición en tiempo de diseño
- Evaluación del bloque de reglas



Pantalla 2

Por medio de los cuadros de dialogo mostrados en la pantalla 2 se puede introducir los parámetros necesarios para generar un controlador básico.

Si se dejan las opciones por defecto, las cuales son 3 variables de entrada, una variable de salida y un bloque de reglas y se presiona el botón de Ok se dibuja en la pantalla de trabajo el diagrama a a bloques del controlador.



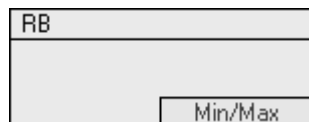
Pantalla 3

En la pantalla 3 se muestra en la pantalla de trabajo:

una representación gráfica para las entradas difusas



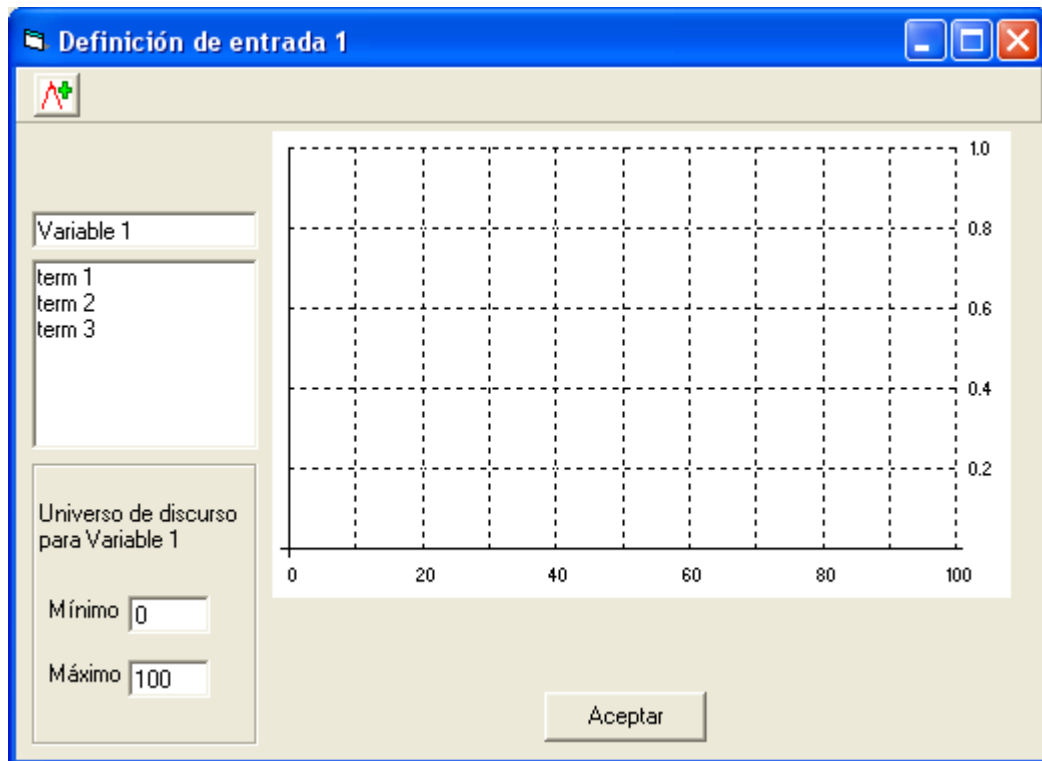
una representación gráfica para el bloque de reglas



una representación gráfica para la salida.



Si se presiona el botón derecho del ratón sobre la figura que representa la entrada aparece la siguiente pantalla:



Pantalla 4

En la pantalla 4 se definen los conjuntos difusos para la entrada seleccionada, en los cuales se puede seleccionar en la parte izquierda de la pantalla los términos de la entrada y renombrarlos, así como agregar más términos a la entrada.

Si se hace doble clic sobre uno de los términos aparece la siguiente pantalla:

Propiedades del término term 1

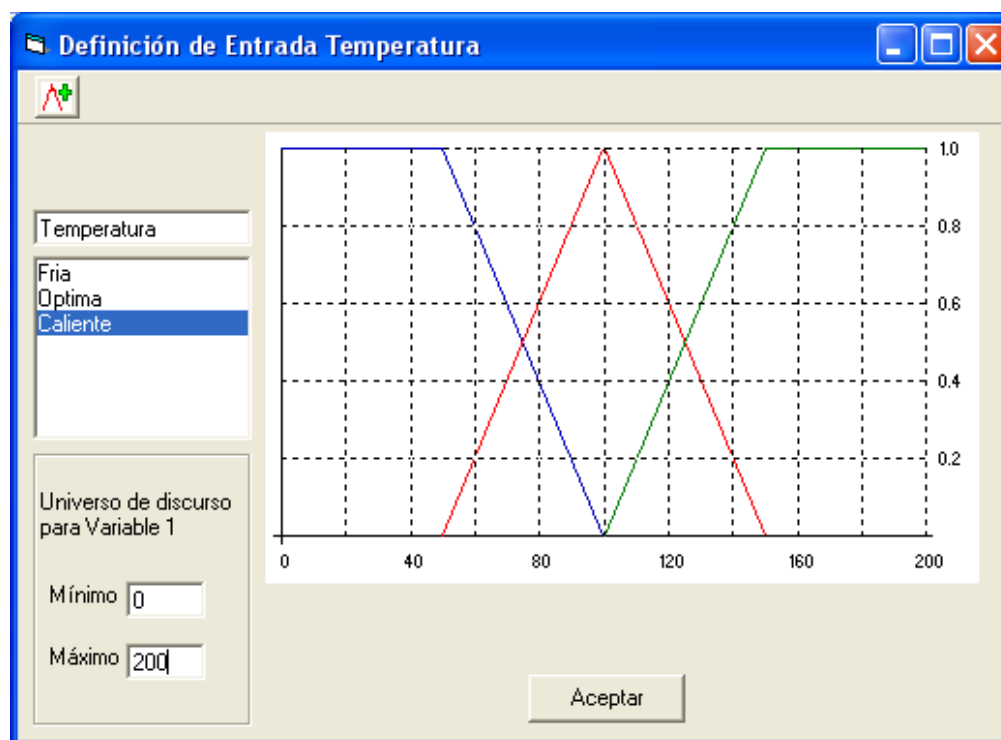
☐ Triangular
☒ Trapezoidal

Nombre del Término:

Punto 1: (0 , 1)
 Punto 2: (0 , 1)
 Punto 3: (25 , 1)
 Punto 4: (50 , 0)

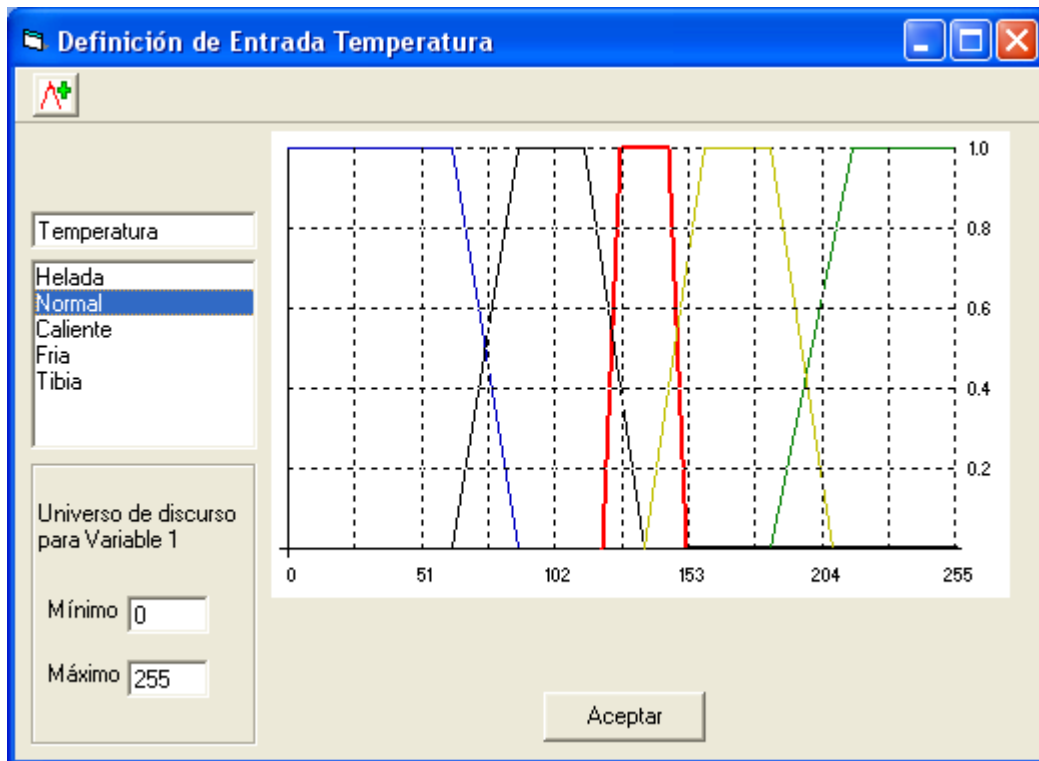
Pantalla 5

En la pantalla 5 se muestra la definición para un conjunto difuso del primer término, el cual por defecto aparece como un conjunto de tipo trapezoidal, en esta pantalla se puede renombrar dicho término y definir los interceptos con el eje de las abscisas; si se selecciona Aceptar aparece una representación gráfica del conjunto:



Pantalla 6

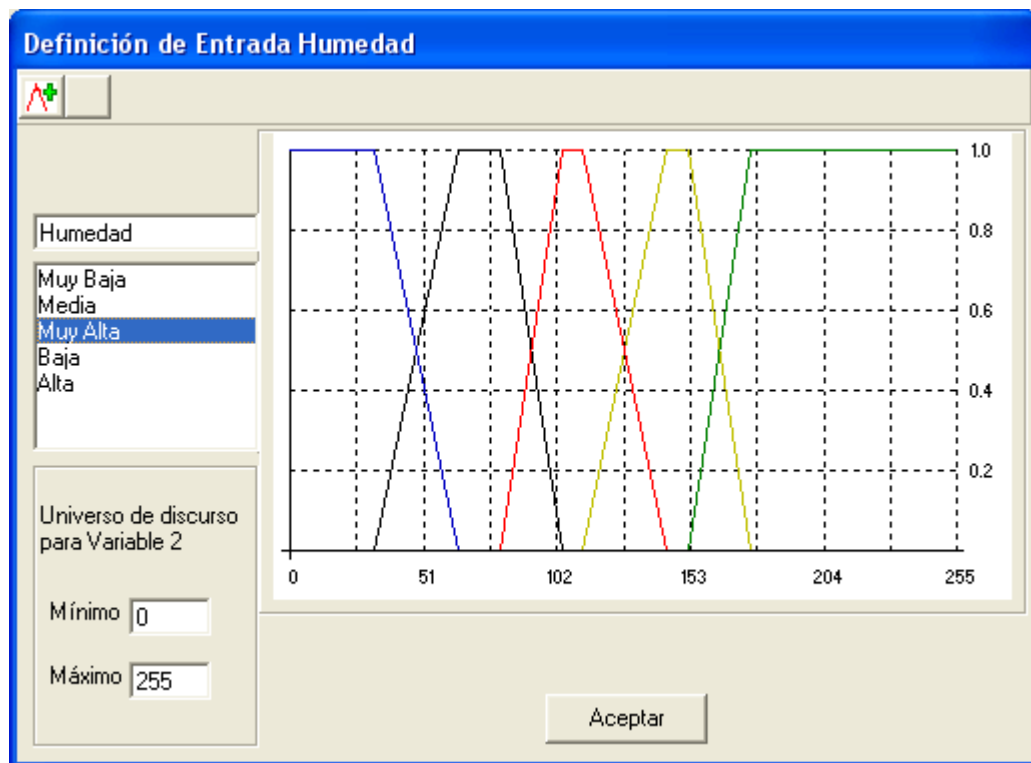
Una vez definidos los términos componentes del conjunto difuso, se obtiene la representación similar a la siguiente:



Pantalla 7

En la parte inferior izquierda se puede definir el Universo de discurso para la variable de entrada, para ello basta posicionar el cursor sobre los campos definidos como Mínimo y Máximo y digitar los valores deseados; en este ejemplo se presentan los valores por defecto: 0 y 100.

Luego se define la segunda entrada:



Pantalla 8

Una vez definidos los conjuntos difusos para las entradas del sistema, se procede a la elaboración de las reglas de inferencia difusa; para ello se hace clic derecho sobre la representación gráfica del bloque de reglas, ante lo cual aparece la siguiente pantalla:

Edición del primer bloque de reglas

Quando

Temperatura

Humedad

Temperatura

Helada

Normal

Caliente

Fria

Tibia

Muy Baja

Media

Muy Alta

Alta

Baja

Humedad

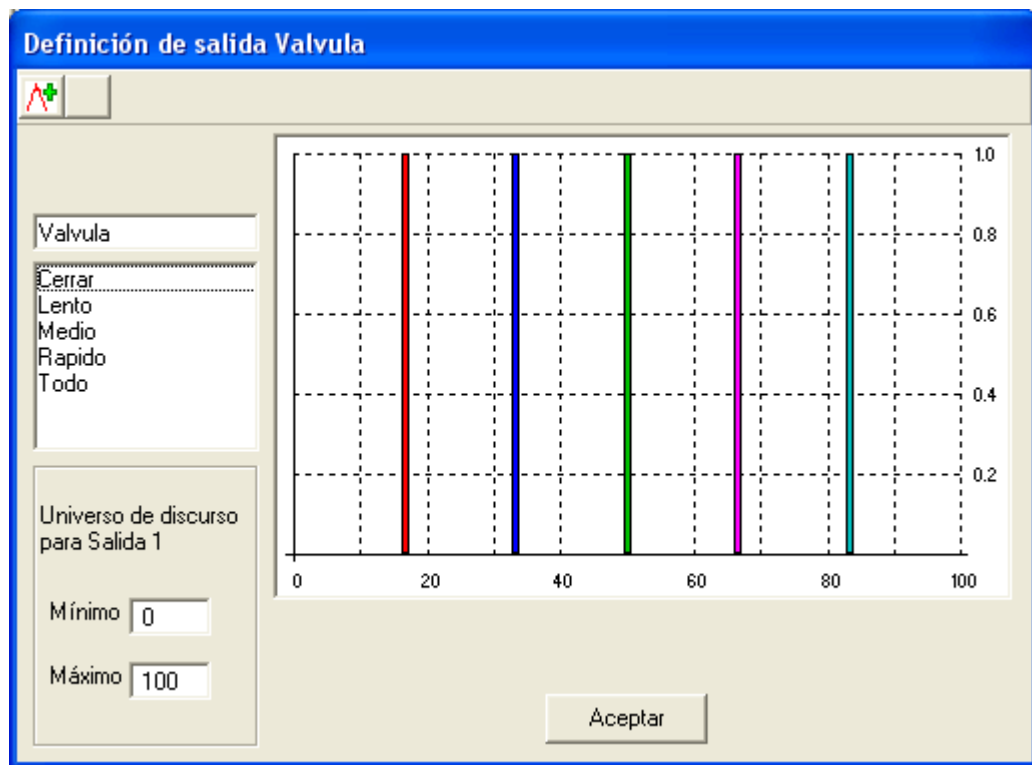
Entonces

Salida 1

Aceptar

Pantalla 9








En la pantalla 9 se muestra el formulario utilizado para la generación de las reglas, en el se introducen los antecedentes y los consecuentes de las reglas en la matriz superior derecha, para poder editarlo debe definirse primero la salida de la siguiente manera:



Pantalla 10

En la pantalla 11 se muestra la definición de las reglas difusas para el presente ejemplo.

Edición del primer bloque de reglas

Cuando
 Temperatura
 Tibia

Humedad
 Muy Baja

Entonces
 Valvula
 Lento

Temperatura



Helada	Cerrar	Cerrar	Cerrar	Cerrar	Cerrar
Normal	Lento	Medio	Medio	Lento	Medio
Caliente	Medio	Rapido	Todo	Rapido	Todo
Fria	Cerrar	Lento	Lento	Cerrar	Lento
Tibia	Lento	Rapido	Rapido	Medio	Rapido

Muy Baja Media Muy Alta Baja Alta

Humedad


Aceptar

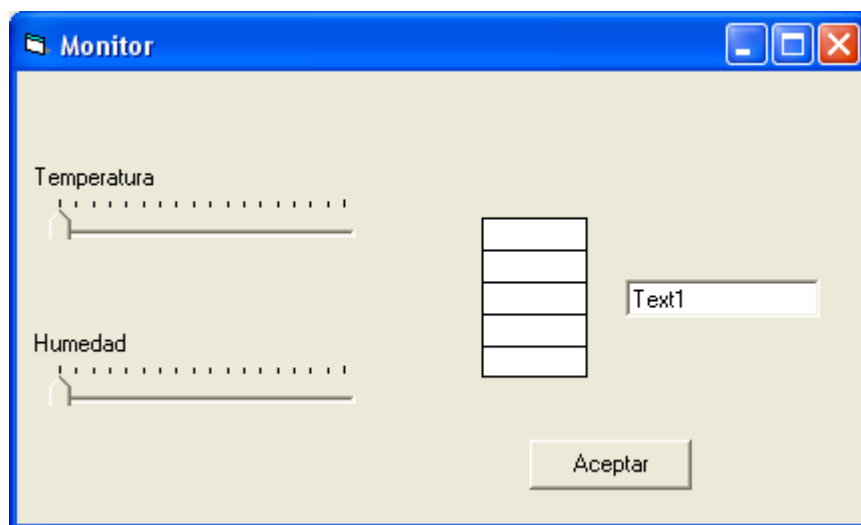
Pantalla 11

Por medio del botón superior izquierdo  se realiza la generación de las reglas, si se presiona posteriormente el botón  se puede obtener de forma textual las difusas

Vista de reglas		
	SI	ENTONCES
Temperatura	Humedad	Valvula
Helada	Muy Baja	Cerrar
Helada	Media	Cerrar
Helada	Muy Alta	Cerrar
Helada	Baja	Cerrar
Helada	Alta	Cerrar
Normal	Muy Baja	Lento
Normal	Media	Medio
Normal	Muy Alta	Medio
Normal	Baja	Lento
Normal	Alta	Medio
Caliente	Muy Baja	Medio
Caliente	Media	Rapido
Caliente	Muy Alta	Todo
Caliente	Baja	Rapido
Caliente	Alta	Todo
Fria	Muy Baja	Cerrar
Fria	Media	Lento
Fria	Muy Alta	Lento
Fria	Baja	Cerrar
Fria	Alta	Lento
Tibia	Muy Baja	Lento
Tibia	Media	Rapido
Tibia	Muy Alta	Rapido

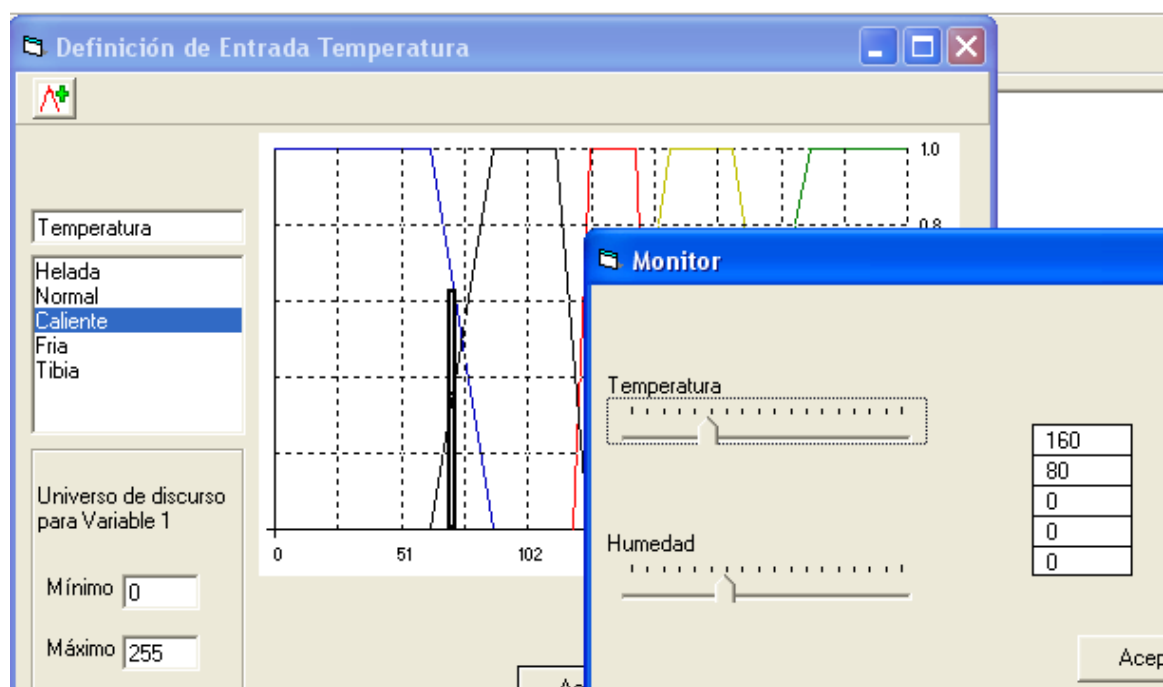
Pantalla 12.

Posteriormente se proporciona el botón  que sirve para obtener una pantalla de simulación según se ilustra a continuación:




Pantalla 13


Por medio de los controles de deslizamiento proporcionados en esta pantalla se puede variar el valor de entrada de cada una de las variables, en la siguiente pantalla se muestra el efecto también sobre las entradas, en cuyas gráficas se puede ver el valor de pertenencia hacia los distintos términos.



Pantalla 14.

Una vez obtenidas las reglas difusas y realizado las pruebas en el simulador se procede a la generación del archivo ASM para descargarlo posteriormente al microcontrolador Motorola 68 HC12 por medio del botón . Con el cual se genera un archivo ASM sobre la base de las características del controlador que ha definido el usuario mediante la adición de pequeños archivos .ASM que contienen segmentos de programa y que son llamados según sea necesario, para completar los datos necesarios para crear el archivo correspondiente a un controlador difuso usando el MCU68HC12, la figura A.4.1 muestra los campos antes mencionados.

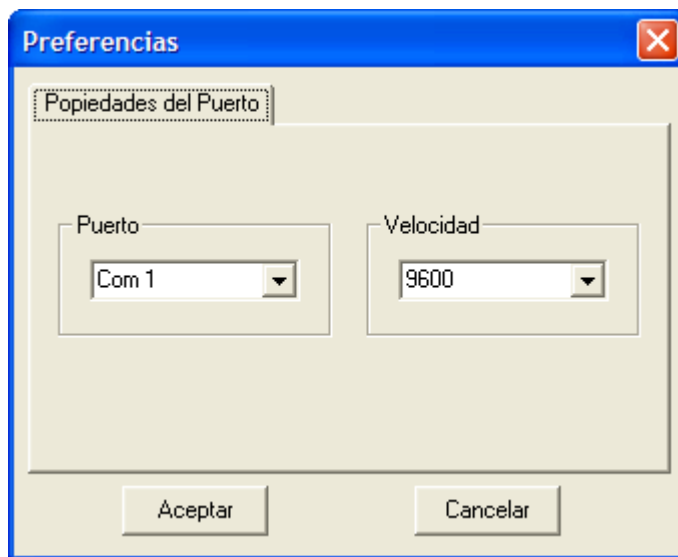
Así en caso que se haya definido 3 entradas el programa agregara el código correspondiente a cada entrada y le escribirá la línea de programa necesaria para evaluar los términos que cada entrada tenga definida, así mismo escribirá las variables necesarias para el controlador y demás campos en la secuencia mostrada en la figura antes mencionada.

Lo anterior se escribe sobre un archivo temporal el cual puede ser guardado haciendo uso del botón  en la barra de herramientas de la pantalla principal para así asignarle un nombre con extensión ASM que permita al usuario poder identificarlo con facilidad y así descargarlo posteriormente al MCU a través del puerto serie.

Definición de variables	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Definición de punteros de memoria	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Definición de términos para las entradas y salida	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Definición de los niveles de activación de los terminos de salida	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Definición de los conjuntos difusos de entrada	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Definición de reglas de inferencia difusa	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Código para evaluar la membresia en N entradas	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Código para evaluación de reglas	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Código para defuzificar la salida	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Código para generar la acción de control	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Rutina de inicialización	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
Rutinas de servicio para N entradas	
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~
~ ~ ~ ~ ~	~ ~ ~ ~ ~

Figura A.4.1 Campos necesarios para crear el archivo ASM correspondiente a un controlador difuso

Dentro del menú principal existe la opción de Herramientas y luego preferencias, en dicha opción se puede configurar el puerto de comunicación la tarjeta del MCU así como la velocidad de la conexión con la misma, según se muestra en la pantalla 15



Pantalla 15

CAPITULO CUATRO. Introducción a la

Lógica Difusa.

Conceptos Básicos de Lógica Difusa¹¹

Hay un número de estrategias de programación en lógica difusa. Este capítulo se concentra en los métodos implementados en las instrucciones de la CPU12 para lógica difusa. La meta primaria es dar un antecedente para una explicación detallada de las instrucciones de lógica difusa de la CPU12.

En general, la lógica difusa proporciona definiciones de conjuntos que tienen límites difusos (fuzzy) en lugar de límites nítidos (crisp) de la lógica de Aristóteles. Estos conjuntos pueden traslaparse de modo que, para un valor específico de entrada, uno o más conjuntos asociados con etiquetas lingüísticas puede ser verdad hasta cierto grado al mismo tiempo. Al variar la entrada desde el rango de un conjunto dentro del rango de un conjunto adyacente, el primer conjunto llega a ser progresivamente menos verdadero mientras que el segundo conjunto llega a ser progresivamente más verdadero.

La lógica difusa tiene funciones de pertenencia que emulan conceptos humanos como “la temperatura es tibia”; esto es, condiciones que son percibidas con límites graduales. Este concepto parece ser un elemento clave de la habilidad humana para resolver ciertos tipos de problemas complejos que han eludido a los métodos tradicionales de control.

Los conjuntos Fuzzy brindan un medio para usar expresiones lingüísticas como “la temperatura es tibia” en reglas que pueden entonces ser evaluadas con un alto grado de

¹¹ Tomado de 1.- NILSSON Nils. J Inteligencia Artificial Una Nueva Síntesis y de 6.- MC68HC12 Advanced Information, Motorola Semiconductors

precisión numérica y repetidamente. Esto contradice directamente la percepción errónea tan común de que la lógica difusa produce resultados aproximados; un conjunto específico de condiciones de entrada siempre produce el mismo resultado, justamente como un sistema de control convencional lo hace.

Un sistema de control de lógica difusa basado en microcontrolador tiene dos partes. La primera parte es un núcleo de inferencia fuzzy el cual es ejecutado periódicamente para determinar las salidas del sistema basados en las entradas actuales del sistema. La segunda parte del sistema es una base de conocimiento la que contiene funciones de pertenencia y Reglas. La figura 1 es un diagrama a bloques de esta clase de sistema de lógica difusa.

La base de conocimiento puede ser desarrollada por un experto en la aplicación sin ninguna experiencia en programación de microcontroladores. Las funciones de pertenencia son simplemente expresiones del entendimiento del experto de los términos lingüísticos que describen al sistema a ser controlado.

Las reglas son postulados en lenguaje ordinario que describen las acciones que un humano experto tomaría para resolver el problema de la aplicación. Las reglas y funciones de pertenencia pueden ser reducidas a estructuras de datos relativamente simples (la base de conocimiento) almacenados en memoria no volátil. Un núcleo de inferencia fuzzy puede ser escrito por un programador quien no conoce como trabaja el sistema de la aplicación. La única cosa que el programador necesita hacer con la información de la base de conocimiento es almacenarla en localidades de memoria usadas por el núcleo.

Un pase de ejecución del núcleo de inferencia fuzzy genera señales de salida del sistema en respuesta a condiciones actuales de entrada. El núcleo es ejecutado tan a menudo como sea necesario para mantener el control. Si el núcleo es ejecutado más a menudo de lo necesario, el

ancho de banda y la potencia del procesador se desperdician; tardándose mucho entre pases puede causar que el sistema se salga de control. La elección de la tasa periódica para un sistema de control fuzzy es lo mismo que sería para un sistema de control convencional.

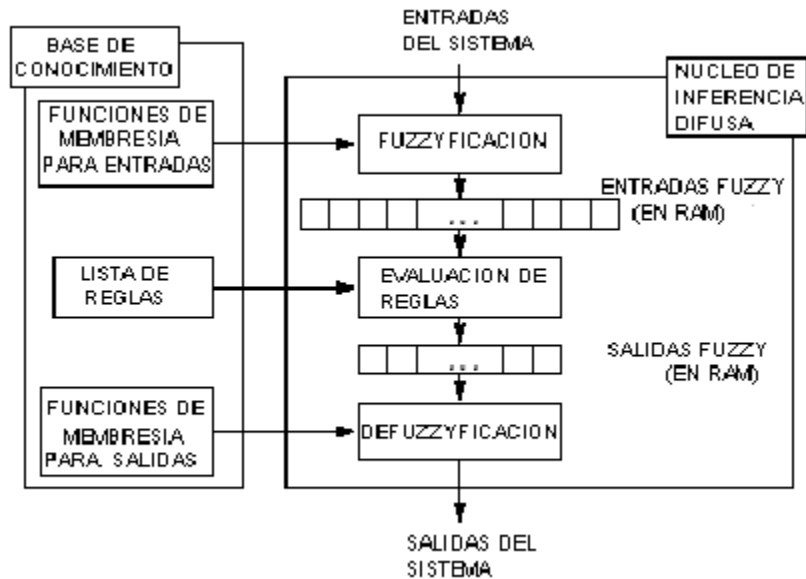


Figura 1 Diagrama a Bloques de un Sistema de Lógica Difusa

4.1 Fuzzyficación (MEM)

Durante el paso de fuzzyficación, los valores de entrada actuales del sistema son comparados contra funciones de pertenencia de entrada almacenadas para determinar el grado que cada etiqueta de entrada del sistema es verdad. Esto se logra hallando el valor y para el valor actual de entrada en una función de pertenencia trapezoidal para cada etiqueta de cada entrada del sistema.

La instrucción MEM en la CPU12 realiza este cálculo para una etiqueta de una entrada del sistema. Para realizar la tarea de fuzzyficación completa para un sistema, varias instrucciones MEM deben ser ejecutadas, usualmente en una estructura de lazo del programa.

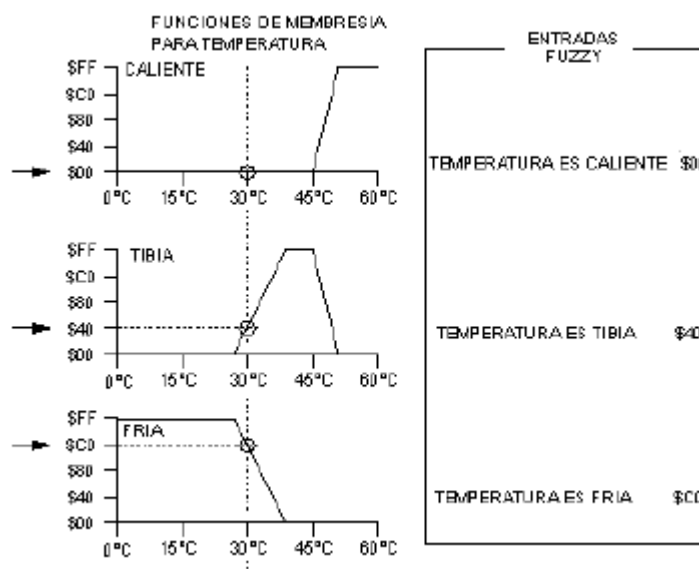


Figura 2

La figura 2 exhibe un sistema de tres funciones de pertenencia de entrada, una para cada etiqueta de la entrada del sistema. El eje x de las tres funciones de pertenencia representa el rango de valores posibles de la entrada del sistema. La línea vertical a través de las tres funciones de pertenencia representa a un valor específico de entrada del sistema. El eje y representa el grado de verdad y varía desde completamente falso (\$00 o 0%) hasta completamente verdad (\$FF o 100%).

El valor y donde la línea vertical intercepta a cada una de las funciones de pertenencia, es el grado al cual el valor actual de entrada concuerda con la etiqueta asociada para esta entrada del sistema. Por ejemplo, la expresión “la temperatura es tibia” es 25% verdad (\$40). El valor \$40 es almacenado en una localidad RAM, y es llamado una entrada difusa (en este

caso, la entrada difusa para “la temperatura es tibia”). Hay una localidad RAM para cada entrada difusa (por cada etiqueta de cada entrada del sistema).

Cuando el paso de fuzzyficación empieza, el valor actual de la entrada del sistema está en un acumulador de la CPU12, un registro índice apunta a la primera definición de función de pertenencia en la base de conocimiento, y un segundo registro índice apunta a la primera entrada difusa en RAM. En cuanto cada entrada difusa es calculada al ejecutar una instrucción MEM, el resultado es almacenado en la entrada difusa y ambos apuntadores son actualizados automáticamente para apuntar a las localidades asociadas con la siguiente entrada difusa. La instrucción MEM se encarga de todo excepto de contar el número de etiquetas por entrada del sistema y de cargar el valor actual de cualquiera entrada subsecuente del sistema.

El resultado final del paso de fuzzyficación es una tabla de entradas difusas representando las condiciones actuales del sistema.

4.2 Evaluación de Reglas (REV y REVW)

La evaluación de reglas es el elemento central de un programa de inferencia de lógica difusa. Este paso procesa una lista de reglas de la base de conocimiento usando valores actuales de entradas difusas en RAM para producir una lista de salidas difusas en RAM. Dichas salidas difusas pueden considerarse como sugerencias algo burdas de lo que la salida del sistema debería ser en respuesta a las actuales condiciones de entrada. Antes que los resultados puedan aplicarse, las salidas difusas deben de procesarse posteriormente, o defuzzyficadas, para producir un solo valor de salida que representa el efecto combinado de todas las salidas difusas.

La CPU12 ofrece dos variaciones de instrucciones para evaluación de reglas. La instrucción REV provee para reglas no ponderada (todas las reglas se consideran de igual importancia). La instrucción REVW es similar pero permite a cada regla tener un factor de peso separado el cual es almacenado en una estructura separada paralela de datos en la base de conocimiento. Aparte de los pesos, las dos instrucciones para evaluación de reglas también difieren en la manera que las reglas son codificadas en la base de conocimiento.

La comprensión de la estructura y la sintaxis de las reglas es necesaria para entender cómo un microcontrolador realiza la tarea de evaluación de reglas. La siguiente es un ejemplo de una regla típica.

Si la temperatura es tibia y la presión es alta entonces el calor es (debería ser) apagado.

A primera vista, parece que codificar esta regla en una forma compacta comprensible para el microcontrolador sería difícil, pero es realmente simple reducir la regla a una lista breve de apuntadores de memoria. La porción izquierda de la regla es un postulado de condiciones de la entrada y la porción derecha de la regla es un postulado de acciones de salida. La porción izquierda de una regla está formada de uno o más (en este caso dos) antecedentes conectados por un operador fuzzy **and**. Cada expresión antecedente consiste del nombre de la entrada del sistema, seguido por **es**, seguido por un nombre de etiqueta. La etiqueta debe estar definida mediante una función de pertenencia en la base de conocimiento. Cada expresión antecedente corresponde a una de las entradas difusas en RAM. Puesto que **y** es el único operador permitido para conectar expresiones antecedentes, no hay necesidad de incluir a éstos en la codificación de la regla. Los antecedentes pueden codificarse como una simple lista de apuntadores hacia (o direcciones de) las entradas difusas a las cuales ellos hacen referencia.

La porción derecha de una regla se forma de una o más (en este caso una) consecuencias. Cada expresión de consecuencia consiste del nombre de una salida del sistema, seguido por es, seguido por un nombre de etiqueta. Cada expresión de consecuencia corresponde a una salida difusa específica en RAM. Consecuencias para una regla pueden codificarse como una simple lista de apuntadores a (o direcciones de) las salidas difusas a las cuales hacen referencia.

Las reglas completas son almacenadas en la base de conocimiento como una lista de apuntadores o direcciones de entradas difusas y salidas difusas. Para que trabaje la lógica de evaluación de reglas, debe haber algunos medios de saber cuales apuntadores se refieren a entradas difusas, y cuáles se refieren a salidas difusas.

También debe haber una manera de saber cuando la última regla del sistema ha sido alcanzada.

Un método de organización es tener un número fijo de reglas con un número específico de antecedentes y consecuencias. Un segundo método, empleado en núcleos de Motorola Freeware M68HC11, es marcar el fin de la lista de reglas con un valor reservado, y usar un bit en los apuntadores para distinguir antecedentes de consecuencias. Un tercer método de organización, usado en la CPU12, es marcar el fin de la lista de reglas con un valor reservado, y separar antecedentes y consecuencias con otro valor reservado. Ello permite cualquier número de reglas, y permite a cada regla tener cualquier número de antecedentes y consecuencias, sujeto a los límites impuestos por la disponibilidad de memoria en el sistema.

Cada regla es evaluada secuencialmente, pero las reglas como grupo son tratadas como si ellas fueran todas evaluadas simultáneamente. Dos operaciones matemáticas toman lugar durante la evaluación de reglas. El operador fuzzy **and** corresponde a la operación matemática de hallar el mínimo y la operación fuzzy **or** corresponde a la operación matemática de hallar el máximo.

La operación fuzzy **and** es usada para conectar antecedentes dentro de una regla. La operación fuzzy **or** está implícita entre reglas sucesivas. Antes de evaluar cualquier Regla, todas las salidas difusas se ponen a cero (lo que significa nada de verdad). Cuando cada regla es evaluada, el más pequeño (el mínimo) antecedente es tomado para ser la verdad global de la regla. Este valor de verdad de la regla es aplicado a cada consecuencia de la regla (al almacenar este valor a la correspondiente salida difusa) a menos que la salida difusa es ya más grande (máxima). Si dos o más reglas afectan a la misma salida difusa, la regla que es más verdadera gobierna el valor en la salida difusa porque las reglas se conectan por un fuzzy **or** implícito.

En el caso de reglas ponderadas, el valor de verdad para una regla se determina del modo usual hallando el más pequeño antecedente de regla. Antes de aplicar este valor de verdad a las consecuencias para la regla, el valor se multiplica por una fracción de cero (regla no habilitada) a uno (regla totalmente habilitada). El resultado modificado del valor de verdad se aplica entonces a las salidas difusas.

El resultado final del paso de evaluación reglas es una tabla de salidas difusas sugeridas o “crudas” en RAM. Estos valores fueron obtenidos al alimentar condiciones actuales (valores de entradas difusas) en las reglas del sistema en la base de conocimiento. Estos resultados crudos no pueden ser suministrados directamente a las salidas del sistema porque ellos pueden ser ambiguos. Por ejemplo, una salida cruda puede indicar que la salida del sistema debería ser media con a grado de verdad de 50% mientras, al mismo tiempo, otra indica que la salida del sistema debería ser baja con a grado de verdad de 25%. El paso de defuzzyficación resuelve tales ambigüedades.

4.3 Defuzzyficación (WAV)

El paso final en el programa de lógica difusa combina las salidas difusas crudas en una salida compuesta del sistema. A diferencia de las formas trapecoide usadas para las entradas, la CPU12 típicamente usa barras (singletons) para funciones de pertenencia de salida. Al igual que con las entradas, el eje x representa el rango de posibles valores para la salida del sistema. Las funciones de pertenencia tipo barra (Singleton) consisten de las posiciones en el eje x para cada etiqueta de la salida del sistema. Las salidas difusas corresponden a la altura en el eje y de la correspondiente función de pertenencia de salida.

La instrucción WAV calcula las sumas del numerador y del denominador para el promedio ponderado de las salida difusas de acuerdo con la fórmula:

$$\text{salida del sistema} = \frac{\sum_{j=1}^n S_j F_j}{\sum_{j=1}^n F_j}$$

Donde n es el número de etiquetas de la salida del sistema, S_j son las posiciones de los singleton de la base de conocimiento, y F_j son las salidas difusas en RAM. Para un programa común de lógica difusa en la CPU12, n es ocho o menos (aunque esta instrucción puede manejar cualquier valor hasta 255) y S_j y F_j son valores de 8 bits . La división final es realizada con una instrucción separada EDIV colocada inmediatamente después de la instrucción WAV.

Antes de ejecutar WAV, un acumulador debe be cargado con el número de iteraciones (n), un registro índice debe apuntar a la lista de posiciones singleton en la base de conocimiento, y un segundo registro índice debe apuntar a la lista de salidas difusas en RAM. Si el sistema tiene

más que una salida del sistema, la instrucción WAV es ejecutada una vez por cada salida del sistema.

CAPITULO 5. Soporte para Lógica

Difusa en el MCU 68HC12

La CPU12 tiene el primer conjunto de instrucciones de un microcontrolador para dirigirse específicamente a las necesidades de la lógica difusa. Esta sección describe el uso de la lógica difusa en sistemas de control, discute las instrucciones de lógica difusa de la CPU12, y proporciona ejemplos de programas de lógica difusa¹².

5.1 Introducción

La CPU12 incluye cuatro instrucciones que realizan tareas específicas de lógica difusa. Además, varias otras instrucciones son especialmente útiles en programas de lógica difusa. La amigabilidad global tipo C del conjunto de instrucciones también ayuda al desarrollo de programas eficientes de lógica difusa.

Esta sección explica lo básico del algoritmo de la lógica difusa para el cual las cuatro instrucciones de lógica difusa están dedicadas. Cada una de las instrucciones de lógica difusa serán explicadas en detalle. Finalmente, otros algoritmos de lógica difusa del cliente son discutidos, con énfasis en el uso de otras instrucciones de la CPU12.

Las cuatro instrucciones de lógica difusa son

- **MEM**, (determina el grado de pertenencia), la cual evalúa funciones de pertenencia trapezoidales;
- **REV** (evaluación de reglas de lógica difusa) y **RE VW** (evaluación ponderada de reglas de lógica difusa), las que realizan evaluación MIN-MAX de reglas no ponderada o ponderada; y

¹² la teoría fue tomada de 6.- MC68HC12 Advanced Information, Motorola Semiconductors, y los ejemplos de la base de conocimiento del tema por parte de los autores

- **WAV**, la que realiza defuzzificación por promedio ponderado con funciones de pertenencia de salida tipo singleton (barra).

Otras instrucciones que son útiles para programas de lógica difusa incluyen:

- **MINA** (coloca el menor de dos valores sin signo de 8 bits en el acumulador A),
- **EMIND** (coloca el menor de dos valores sin signo de 16 bits en el acumulador D),
- **MAXM** (coloca el mayor de dos valores sin signo de 8 bits en memoria),
- **EMAXM** (coloca el mayor de dos valores sin signo de 16 bits en memoria),
- **TBL** (Búsqueda en tablas e interpolación),
- **ETBL** (Búsqueda extendida en tablas e interpolación), y
- **EMACS** (Multiplicación y Acumulación extendida, con signo; 16*16 bits para obtener 32 bits).

Para mayor resolución de programas fuzzy, las instrucciones matemáticas rápidas de precisión extendida de la CPU12 son también benéficas. Los modos de direccionamiento indexado flexibles ayudan a simplificar el acceso de la lógica difusa a estructuras de datos almacenados como listas o tablas como estructuras en memoria.

Las adiciones a la lógica actual requeridas para implementar el soporte a la lógica difusa en la CPU12 son muy pequeñas, de modo que no hay incremento apreciable en costo para el usuario típico. Un núcleo de inferencia fuzzy para la CPU12 requiere un quinto a lo más de espacio de código, y se ejecuta quince veces más rápido que un núcleo comparable implementado en un microcontrolador típico de rango medio.

Al incorporar el soporte de lógica difusa a una familia de productos de microcontroladores de propósito general, Motorola ha hecho que la lógica difusa esté

disponible para una enorme base de aplicaciones.

5.2 Ejemplo de un Núcleo de Inferencia

01	[2]	FUZZYFICA	LDX	#FMS_ENT	; Apunta a definiciones de FM
02	[2]		LDY	#ENT_FUZ	; Apunta a tabla de entradas fuzzy
03	[3]		LDAA	VALOR_ACT	; Trae 1er valor de entrada
04	[1]	LAZO1	LDAB	#7	; 7 etiquetas por entrada
05	[5]		MEM		; Evalúa una FM
06	[3]		DBNE	B,LAZO1	; Para 7 etiquetas por entrada
07	[3]	LAZO2	LDAA	VALOR_ACT+1	; Trae 2do valor de entrada
08	[1]		LDAB	#7	; 7 etiquetas por entrada
09	[5]		MEM		; Evalúa una FM
10	[3]		DBNE	B,LAZO2	; Para 7 etiquetas por entrada
11	[1]	EVALUA	LDAB	#7	; contador del lazo
12	[2]		CLR	1,Y+	; Limpia sal. Fuzzy e incr. Apunt.
13	[3]		DBNE	B,EVALUA	; hasta limpiar todas las sal. Fuzzy
14	[2]		LDX	#REGLA1	; Apunta a 1er elemento de reglas
15	[2]		LDY	#ENT_FUZ	; Apunta a ent y sal. fuzzy
16	[1]		LDAA	#\$FF	; Inicia A (y limpia bit V)
17	[3n+4]		REV		; Procesa lista de reglas
18	[2]	DEFUZ	LDY	#SAL_FUZ	; Apunta a salidas fuzzy
19	[1]		LDX	#POS_SGLTN	; Apunta a posiciones singleton
20	[1]		LDAB	#7	; 7 salidas fuzzy por salida crisp
21	[8b+9]		WAV		; Hace suma por prom ponderado
22	[11]		EDIV		; Divide por suma ponderada
23	[1]		TFR	Y, D	; Pone resultado en A:B
24	[3]		STAB	SAL_COG	; Almacena salida del sistema

Figura 1

La figura 1 es un núcleo completo de inferencia fuzzy escrito en lenguaje ensamblador de la CPU12. Los números en paréntesis cuadrados son cuentas de ciclos. El núcleo usa dos entradas al sistema con siete etiquetas cada una y una salida del sistema con siete etiquetas. El programa se ensambla con 57 bytes. Y se ejecuta en cerca de 54 ms a una tasa de bus de 8 MHz. La estructura básica puede fácilmente extenderse a un sistema de propósito general con mayor número de entradas y salidas.

Las líneas 1 a 3 preparan los apuntadores y cargan el valor de la entrada al sistema en el acumulador A.

La línea 4 fija el contador de lazo para el lazo en las líneas 5 y 6.

Las líneas 5 y 6 efectúan el lazo de fuzzificación para las siete etiquetas de una entrada al sistema. La instrucción MEM halla el y valor en una función de pertenencia trapezoidal para el valor actual entrada, para una etiqueta de la entrada actual, y entonces salva el resultado a la correspondiente entrada difusa. Los apuntadores en X y Y son automáticamente actualizados cuatro y una localidades de modo que ellos apuntan a la siguiente función de pertenencia y entrada difusa respectivamente.

La línea 7 carga el valor actual de la siguiente entrada del sistema. Los apuntadores en X y Y ya apuntan a los lugares correctos como resultado de la función de actualización automática de la instrucción MEM en la línea 5.

La línea 8 recarga el contador de lazo.

Las líneas 9 y 10 forman un lazo para fuzzyficar las siete etiquetas de la segunda entrada del sistema.

Cuando el programa llega a la línea 11, el registro índice Y está apuntando a la siguiente localidad después de la última entrada difusa, la cual es la primera salida difusa en este sistema.

La línea 11 fija el contador de lazo para limpiar las siete salidas difusas.

Las líneas 12 y 13 forman un lazo para limpiar todas las salidas difusas antes de que la evaluación de reglas arranque.

La línea 14 inicia al registro índice X para apuntar al primer elemento en la lista de reglas para la instrucción REV.

La línea 15 inicia al registro índice Y para apuntar a las entradas y salidas difusas en el sistema. La lista de reglas (para REV) consiste de sesgos de 8 bits desde esta dirección base a

entradas difusas o salidas difusas particulares. El valor especial \$FE se interpreta por REV como un marcador entre antecedentes y consecuencias de las reglas.

La línea 16 inicia al acumulador A al valor más alto de 8 bits en preparación para hallar la entrada difusa más pequeña referenciada por el antecedente de la regla. La instrucción LDAA #\$FF también limpia el bit V en el registro de códigos de condición de la CPU12 de modo que la instrucción REV sabe que está procesando antecedentes.

Durante el proceso de la lista de reglas, El bit V es conmutado cada vez que un \$FE es detectado en la lista. El bit V indica cuando REV está procesando antecedentes o consecuencias.

La línea 17 es la instrucción REV, un lazo auto contenido para procesar elementos sucesivos en la lista de reglas hasta que un carácter \$FF aparece. Para un sistema de 17 reglas con dos antecedentes y una consecuencia para cada una, la instrucción REV toma 259 ciclos, pero ella puede ser interrumpida de modo que no causa una latencia larga en la interrupción.

Las líneas 18 a 20 preparan los apuntadores y un contador de iteraciones para la instrucción WAV.

La línea 21 es el comienzo de la defuzzyficación. La instrucción WAV calcula una suma de productos y una suma de pesos.

La línea 22 completa la defuzzyficación. La instrucción EDIV realiza una división de 32 bits por 16 bits en los resultados intermedios de la WAV para obtener el promedio ponderado.

La línea 23 mueve el resultado de EDIV al acumulador doble.

La línea 24 salva los 8 bits bajos del resultado de la defuzzyficación.

Este ejemplo de programa de inferencia muestra lo fácil que es incorporar la lógica difusa en aplicaciones generales usando la CPU12. El espacio del código y el tiempo de ejecución no son más factores serios en la decisión para usar lógica difusa.

La siguiente sección ofrece más detalles de la instrucción de lógica difusa MEM de la CPU12.

5.3 Detalles de la Instrucción MEM

Esta sección proporciona una explicación más detallada de la evaluación de una función de pertenencia por la instrucción (MEM), incluyendo detalles acerca de casos anormales especiales para funciones de pertenencia impropriadamente definidas.

5.4 Definición de una Función de Pertenencia para MEM

La figura 4 muestra cómo una función de pertenencia normal es especificada en la CPU12. Típicamente una herramienta de software se usa para entrar funciones de pertenencia gráficamente, y la herramienta genera las estructuras de datos para el procesador y el núcleo de software que se van a usar. Alternativamente, puntos y pendientes para las funciones de pertenencia pueden ser determinados y salvados en memoria con directivas para definir constantes del ensamblador.

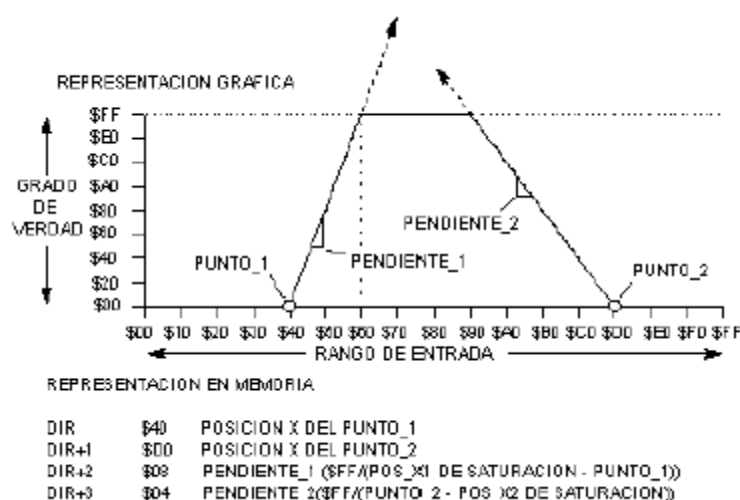


Figura 4 Definición de una Función Normal de Pertenencia

Un algoritmo interno de la CPU calcula el valor y donde la entrada actual intercepta a la función de pertenencia. Este algoritmo asume que la función de pertenencia obedece al sentido común. Si la definición de la función de pertenencia es impropia, los resultados pueden ser inusuales. El punto: 4.2 Definiciones Anormales de Funciones de Pertenencia, discute tales casos. Las siguientes reglas se aplican para funciones de pertenencia normales.

- $\$00 \leq \text{punto_1} < \FF
- $\$00 < \text{punto_2} \leq \FF
- $\text{punto_1} < \text{punto_2}$
- Los lados inclinados del trapecioide se encuentran en, o arriba de, $\$FF$. Cada entrada del sistema como temperatura tiene varias etiquetas como fría, fresca, normal, tibia, y caliente. Cada etiqueta de cada entrada del sistema debe tener una función de pertenencia para describir su significado de una manera numérica no ambigua. Típicamente, hay de tres a siete etiquetas por entrada del sistema, pero no existe restricción práctica sobre este número en cuanto al paso de fuzzyficación concierne.

5.5 Definiciones Anormales de Funciones de Pertenencia

En la CPU12, es posible (y apropiado) definir funciones de pertenencia nítidas (crisp). Una función de pertenencia **crisp** tiene uno o ambos lados vertical (pendiente infinita). Puesto que el valor de pendiente \$00 no se usa de otra manera, él es asignado para indicar pendiente infinita para la instrucción MEM en la CPU12. A pesar de que una buena herramienta de desarrollo fuzzy no permitiría al usuario especificar una función de pertenencia impropia, es posible tener errores de programación o errores de memoria que resultan en funciones de pertenencia erróneas anormales. Aunque estas formas anormales no corresponden a cualquier sistema normal, comprendiendo cómo la CPU12 trata estos casos puede ser útil para depurar.

Un examen más cercano del algoritmo de la instrucción MEM muestra como tales funciones de pertenencia son evaluadas. La figura 5 es un diagrama de flujo completo de la ejecución de la instrucción MEM. Cada caja rectangular representa un ciclo de bus de la CPU. El número en la esquina superior izquierda corresponde al número de ciclo y la letra corresponde al tipo de ciclo (R para lectura, W para escritura). La porción superior de la caja incluye información acerca de la actividad del bus durante este ciclo (sí hay).

La porción inferior de la caja, la cual está separada por una línea punteada, incluye información acerca del proceso interno de la CPU. Es común que varias funciones internas se efectúen durante un solo ciclo de la CPU (por ejemplo, en el ciclo 2, dos sustracciones de 8 bits se llevan a cabo y una bandera es fijada basada en los resultados).

Considerar 4a: Si (((Pendiente_2 = 0) o (Grado_2 > \$FF)) y (flag_d12n = 0)). La bandera flag_d12n es cero siempre que el valor de la entrada (en el acumulador A) esté dentro del trapezoide. En todo lugar fuera del trapezoide, uno o el otro término delta será negativo, y la bandera será uno. La Pendiente_2 igual a cero indica que el lado derecho del trapezoide tiene

pendiente infinita, de modo que el grado resultante debe ser \$FF en todo lugar dentro del trapecioide, incluyendo al punto_2, por lo que a este lado concierne. El término grado_2 mayor que \$FF significa que el valor está lo suficientemente dentro del trapecioide que el lado empinado derecho del trapecioide ha cruzado por arriba del nivel de corte \$FF y el grado resultante debe ser \$FF por lo que al lado derecho empinado concierne. 4a decide si el valor está a la izquierda del lado derecho empinado (Grado = \$FF), o en la porción empinada del lado derecho del trapecioide (Grado = Grado_2). 4b Pudiera todavía cancelar este valor tentativo de Grado.

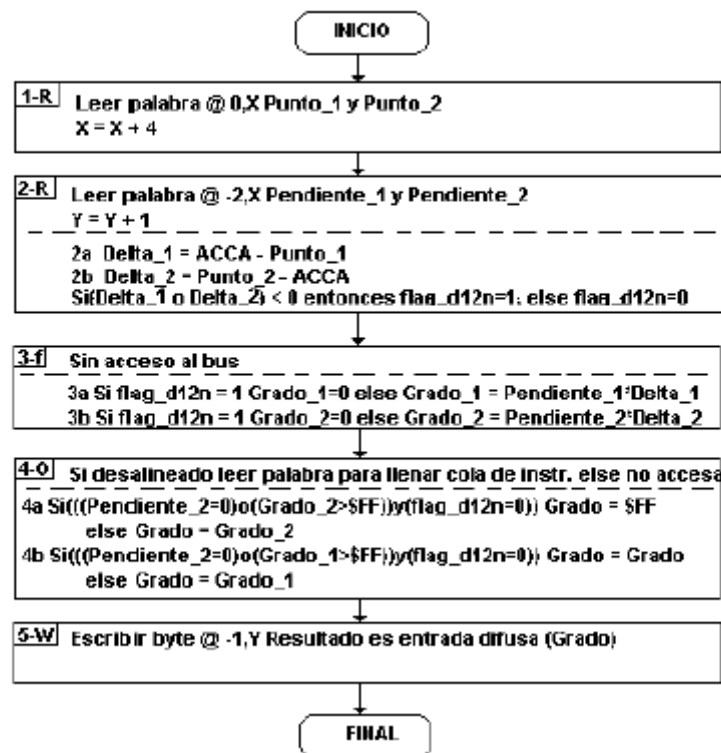


Figura 5 Diagrama de Flujo de la Instrucción MEM

En 4b, la pendiente_1 es cero si el lado izquierdo del trapecioide tiene pendiente infinita (vertical). Si es así, el resultado (grado) debe ser \$FF en y la derecha del punto_1 en todo lugar dentro del trapecioide por lo que al lado izquierdo concierne. El término grado_1 mayor que

\$FF corresponde a la entrada estando a la derecha de donde el lado empinado izquierdo pasa el nivel de corte \$FF. Si cualquiera de estas condiciones es verdad, el resultado (grado) se deja en el valor que obtuvo en 4a. La condición “else” en 4b corresponde a la entrada situada en la porción empinada del lado izquierdo del trapecioide (o posiblemente afuera del trapecioide), así que el resultado es grado igual a grado_1. Si la entrada está afuera del trapecioide, la bandera flag_d12n será uno y grado_1 y grado_2 son forzadas a ser \$00 en el ciclo 3. La condición **else** de 4b fijará el resultado a \$00.

Los siguientes casos especiales representan definiciones anormales de funciones de pertenencia. Las explicaciones describen cómo el algoritmo específico en la CPU12 resuelve estos casos no usuales. Los resultados no son todos intuitivamente obvios, sino que quedan fuera del algoritmo específico. Recordar, tales casos no deberán ocurrir en un sistema normal.

5.5.1 Función de Pertenencia Anormal Caso 1

Esta función de pertenencia es anormal porque los lados empinados se cruzan debajo del nivel de corte \$FF. La bandera flag_d12n fuerza a la función de pertenencia a valer \$00 en todo lugar excepto del punto_1 al punto_2. Dentro de este intervalo, los valores tentativos para grado_1 y grado_2 calculados en el ciclo 3 quedan en los lados empinados que se cruzan. En el paso 4a, el grado es puesto al valor grado_2, pero en 4b esto es invalidado por el valor de grado_1, que termina como el resultado de la instrucción MEM. Una manera de decir esto es que el resultado sigue al lado empinado izquierdo hasta que la entrada pasa el punto_2, donde el resultado se hace \$00.

Definición en Memoria: \$60, \$80, \$04, \$04; Punto_1, Punto_2, Pendiente_1, Pendiente_2

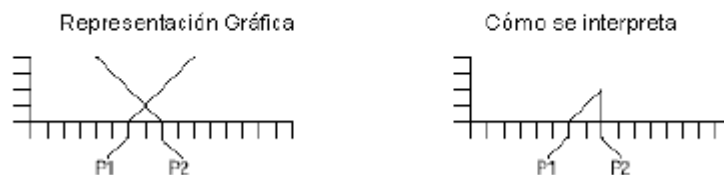


Figura 6 Función de Pertenencia Anormal Caso 1

Si el punto_1 estuviera a la derecha del punto_2, flag_d12n forzaría el resultado a ser \$00 para todos los valores de entrada. De hecho, flag_d12n siempre limita la región de interés al espacio mayor o igual al punto_1 y menor o igual que el punto_2.

5.5.2 Función de Pertenencia Anormal Caso 2

Como el ejemplo previo, la función de pertenencia en el caso 2 es anormal porque los lados empinados se cruzan debajo del nivel de corte \$FF, pero el lado empinado izquierdo llega al nivel de corte \$FF antes que la entrada llegue al punto_2.

En este caso, el resultado sigue al lado empinado izquierdo hasta que éste llega al nivel de corte \$FF. En este punto, el término $(\text{grado}_1 > \$FF)$ de 4b entra en juego, haciendo que la expresión sea verdad de modo que grado se iguala a grado (no invalida). El resultado de aquí al punto_2 llega a ser controlado por la parte “else” de 4a ($\text{grado} = \text{grado}_2$), y el resultado sigue al lado empinado derecho.

Definición en Memoria: \$60, \$C0, \$04, \$04; Punto_1, Punto_2, Pendiente_1, Pendiente_2

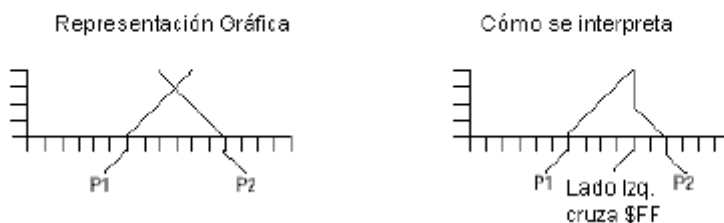


Figura 7 Función de Pertenencia Anormal Caso 2

5.5.3 Función de Pertenencia Anormal Caso 3

La función de pertenencia en el caso 3 es anormal porque los lados empinados se cruzan debajo del nivel de corte \$FF, y el lado empinado izquierdo tiene pendiente infinita. En este caso, 4a no es verdad, así que grado se iguala a grado_2. 4b es verdad porque pendiente_1 es cero, de modo que 4b no invalida a grado.

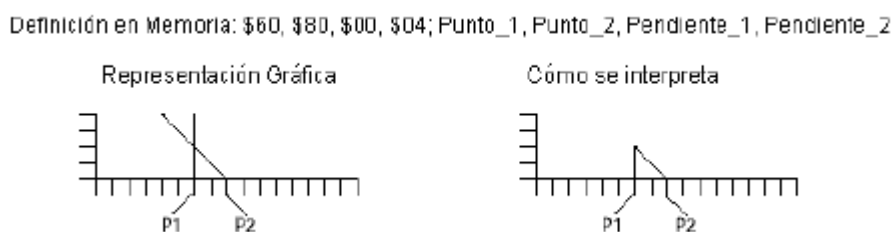


Figura 8 Función de Pertenencia Anormal Caso 3

5.6 Detalles de las Instrucciones REV y REVW

Esta sección proporciona una explicación más detallada de las instrucciones (REV y REVW) para evaluación de reglas. Las estructuras de datos usadas para especificar reglas son algo diferentes para las versiones ponderada versus la no ponderada de la instrucción. Una usa sesgos de 8 bits en las reglas codificadas, mientras que la otra usa direcciones completas de 16 bits. Esto afecta el tamaño de la estructura de datos de la regla y el tiempo de ejecución.

5.6.1 Instrucción REV para Evaluación no Ponderada de Reglas

Esta instrucción implementa evaluación básica min-max de reglas. Los registros de la CPU son usados como apuntadores y para resultados intermedios de los cálculos.

Puesto que la instrucción REV es esencialmente una instrucción para procesar listas, el tiempo de ejecución es dependiente del número de elementos en la lista de reglas. La instrucción REV

es interrumpible (típicamente en tres ciclos del bus), de modo que no afecta de manera adversa en el peor caso la latencia de la interrupción. Ya que todos los resultados intermedios y el estado de la instrucción se ponen en registros de la CPU que son apilados, el código de servicio de la interrupción puede incluso incluir instrucciones REV y REVW independientes.

5.6.2 Preparación Previa a la Ejecución de REV

Algunos registros de la CPU y ciertas localidades de memoria necesitan ser iniciadas justo antes de la ejecución de la instrucción REV. Los registros índice X y Y se usan como índices apuntadores a la lista de reglas y las entradas y salidas difusas. El acumulador A se usa para resultados intermedios de los cálculos y necesita ser puesto a \$FF inicialmente. El bit de código de condición V se usa como un indicador de estado de la instrucción para mostrar si los antecedentes o las consecuencias están siendo procesadas.

Inicialmente, el bit V es puesto a cero para indicar que los antecedentes están siendo procesados. Las salidas difusas (localidades de trabajo en RAM) necesitan ser puestas a \$00. Si estos valores no son iniciados antes de la ejecución de la instrucción REV, los resultados serán erróneos.

El registro índice X se pone a la dirección del primer elemento en la lista de reglas (en la base de conocimiento). La instrucción REV automáticamente actualiza este apuntador de modo que la instrucción puede reasumir correctamente si ella es interrumpida. Después que la instrucción REV termina, X apuntará a la siguiente dirección pasado el caracter separador \$FF que marca el final de la lista de reglas.

El registro índice se pone a la dirección para las entradas y salidas difusas (en RAM de trabajo). Cada antecedente de regla es un sesgo no signado de 8 bits desde la dirección base a

la entrada difusa referida. Cada consecuencia de regla es un sesgo sin signo de 8 bits desde la dirección base a la salida difusa referida. El registro índice permanece constante durante la ejecución de la instrucción REV.

El acumulador A de 8 bits se usa para contener resultados intermedios de los cálculos durante la ejecución de la instrucción REV. Durante el proceso de antecedentes, A empieza con \$FF y es reemplazado por cualquier entrada difusa más pequeña que es referida por un antecedente de regla (MIN). Durante el proceso de consecuencias, A contiene el valor de verdad para la regla. Este valor de verdad es salvado en cualquier salida difusa que es referida por una consecuencia de regla, a menos que la salida difusa sea ya mayor (MAX).

Antes de empezar a ejecutar a REV, A debe ser puesto a \$FF (el valor más grande de 8 bits) porque la evaluación de reglas siempre inicia con el proceso de los antecedentes de la primera regla. Para reglas subsecuentes en la lista, A se pone automáticamente a \$FF cuando la instrucción detecta al carácter marcador \$FE entre la última consecuencia de la regla previa, y el primer antecedente de una nueva regla.

La instrucción LDAA #\$FF limpia al bit V al mismo tiempo que inicia A con \$FF.

Esto satisface el requisito al preparar REV de limpiar el bit V y también el requisito de iniciar A con \$FF. Una vez que la instrucción REV arranca, el valor en el bit V es automáticamente mantenido mientras que los caracteres separadores \$FE son detectados.

El requisito final de limpiar todas las salidas difusas con \$00 es parte del algoritmo MAX. Cada vez que una consecuencia de regla hace referencia a una salida difusa, tal salida difusa es comparada con el valor de verdad de la regla actual. Si el actual valor de verdad es mayor, será escrito encima del valor previo en la salida difusa. Después que todas las reglas han sido

evaluadas, la salida difusa contiene el valor de verdad para la regla más verdadera que hizo referencia a tal salida difusa.

Tras que REV termina, A contendrá el valor de verdad para la última regla en la lista de reglas. El bit de código de condición V debe ser uno porque el último elemento antes del marcador \$FF final debe haber sido una consecuencia de regla. Si V es cero después de la ejecución de REV, ello indica que la lista de reglas fue estructurada incorrectamente.

5.6.3 Interrupción de la Instrucción REV

La instrucción REV incluye un lazo de tres ciclos de procesamiento por cada byte en la lista de reglas (incluyendo antecedentes, consecuencias, y caracteres especiales de separación). Dentro de este lazo, se realiza una inspección para ver si cualquier petición de interrupción válida está pendiente. Si una interrupción es detectada, los registros actuales de la CPU son apilados y la interrupción es servida. Cuando la rutina de servicio de la interrupción finaliza, una instrucción RTI causa que la CPU recupere su contexto previo desde la pila, y la instrucción REV es reanudada como si no hubiera existido la interrupción.

El valor apilado del contador de programa (PC), en el caso de una interrupción a la instrucción REV, apunta a la instrucción REV en lugar de la instrucción que sigue a continuación. Esto causa que la CPU trate de ejecutar una nueva instrucción REV al retornar de la interrupción. Puesto que los registros de la CPU (incluyendo al bit V en el registro de códigos de condición) indican el estado actual de la instrucción REV interrumpida, esto causa que la operación de evaluación de reglas se reanude desde donde debe ser.

5.7 Ejemplo de un CLD con el 68HC12

Para clarificar todo lo anterior se presenta un ejemplo con dos variables de entrada temperatura y humedad, cada una de ellas con una partición de tamaño cinco, y una variable de salida, con partición de cinco también, velocidad de un rociador; para dar un ejemplo de diseño de un sistema de CLD. Se usó el código dado en la figura 4, cambiando el 7 por 5 para adecuarse a los datos. El núcleo de inferencia toma 70.25 ms en ejecutarse.

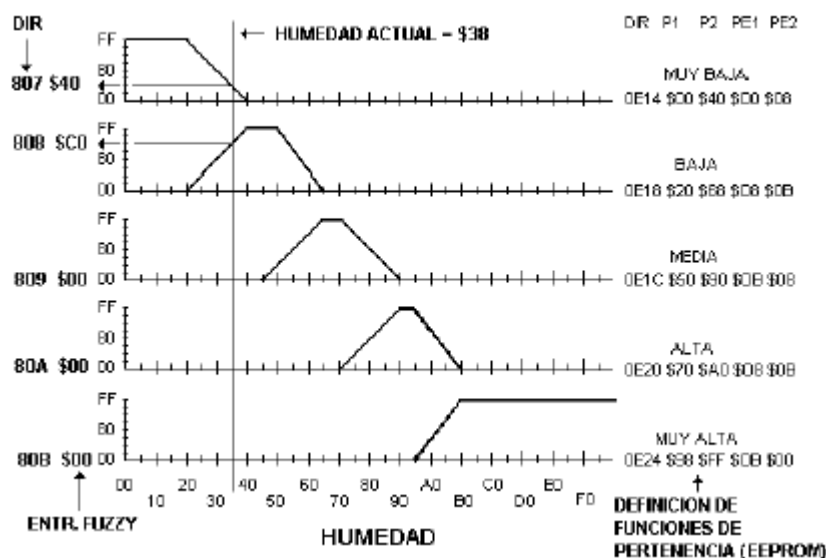


Fig. 10 Proceso de Fuzzyfización para 2da entrada.

La figura 10 muestra el proceso de fuzzyfización de la variable humedad donde el convertidor A/D entrega un valor crisp de \$38 y se generan los cinco valores difusos \$40, \$C0, \$00, \$00 y \$00 para las cinco etiquetas “Muy Baja”, “Baja”, “Media”, “Alta” y “Muy Alta”. El valor crisp de entrada está en la localidad \$801 y las entradas fuzzy generadas quedan de la \$807 a la \$80B, en memoria RAM.

Notar que la definición de las funciones de pertenencia para la humedad queda en memoria EEPROM de la localidad \$0E14 a la \$0E27, esto es cuatro bytes por FP. Todo esto se hace patente al analizar el listado fuente que se ofrece más adelante.

La definición de las FP para la entrada temperatura queda de la \$0E00 a la \$0E13 en EEPROM, según el listado. Las entradas difusas para la temperatura se salvan de la localidad \$802 a la \$806, en RAM. El valor crisp de la temperatura está en la \$800 y se asume que vale \$C8.

Como se explicó en el punto 2 y se abundó en el punto 3, la parte fundamental del diseño de un CLD es la composición de las reglas. La tabla 2 ilustra la matriz de reglas a la que se llega después de un proceso de prueba y error para lograr que el SCLD funcione de manera adecuada.

Temp. → Hum. ↓	Helada \$00	Fria \$00	Normal \$00	Tibia \$58	Cal. \$80
MuyAlta \$00	Cerrar [5]	Lento [10]	Medio [15]	Rapido [20]	Todo [25]
Alta \$00	Cerrar [4]	Lento [9]	Medio [14]	Rapido [19]	Todo [24]
Media \$00	Cerrar [3]	Lento [8]	Medio [13]	Rapido [18]	Rapido [23]
Baja \$C0	Cerrar [2]	Cerrar [7]	Lento [12]	Medio [17]	Rapido [22]
MuyBaja \$40	Cerrar [1]	Cerrar [6]	Lento [11]	Lento [16]	Medio [21]

Tabla 2 Matriz de Reglas de Control.

Como se aprecia, se disparan cuatro reglas: la 16, la 17, la 21 y la 22. La fuerza (verdad) del disparo es el valor mínimo de las verdades que intervienen para disparar tal regla. Estos valores son \$40, \$58, \$40 y \$80, respectivamente.

Las cinco salidas difusas son: Cerrar = \$00, Lento = \$40, Medio = \$58, Rápido = \$80 y Todo = \$00, mismas que se salvan de la localidad \$80C a la \$810, en RAM. Notar que cuando una consecuencia se repite, como en este caso pasa con Medio, la verdad de la salida difusa es la máxima de las verdades, en este caso \$58.

Las FP de salida son del tipo barra (singleton) y están definidas en EEPROM como sigue:

Parar = 00, Lento = \$30, Medio = \$60, Rápido = \$A0 y Todo = \$FF. La instrucción WAV se encarga de calcular el numerador y el denominador de (14) para producir:

$$\begin{aligned}
 &= \frac{\sum_{i=1}^5 S_i \cdot F_i}{\sum_{i=1}^5 F_i} \\
 &= \frac{\$30 \times \$40 + \$60 \times \$58 + \$A0 \times \$80}{\$40 + \$58 + \$80} = \frac{\$7D00}{\$118}
 \end{aligned}$$

Salida del sistema =

Finalmente, la instrucción EDIV genera la señal de control (acción): \$72, que se salva en la \$811.

(c) COPYRIGHT MOTOROLA 1991-1998

Abs. Loc Obj. code Source line

```

-----
1      *****
2      * Programa del Núcleo de Inferencia de Lógica Difusa
3      * un pequeño ejemplo ilustrativo
4      * 2 entradas al sistema con 5 etiquetas CDU
5      * 1 salida del sistema con 5 etiquetas
6      *****
7      ORG $800 ; Empieza memoria de datos (RAM)
8      *****
9      * Las siguientes son las localidades para salvar datos en RAM al calcular
10     *****
11     * entradas al sistema (1 byte cada una en RAM)
12     a000800 Temperatura RMB 1
13     a000801 Humedad RMB 1
14
15     * entradas Fuzzy (1 byte de RAM por etiqueta de entrada del sistema)
16     0000 0802 EntFuzzy: EQU * ;Usada para referirse a entr/sal fuzzy
17     * Temperatura
18     a000802 Helada RMB 1
19     a000803 Fria RMB 1

```

20 a000804 Normal RMB 1
 21 a000805 Tibia RMB 1
 22 a000806 Caliente RMB 1
 23 * Humedad
 24 a000807 MuyBaja RMB 1
 25 a000808 Baja RMB 1
 26 a000809 Media RMB 1
 27 a00080A Alta RMB 1
 28 a00080B MuyAlta RMB 1
 29 * Salidas Fuzzy (1 byte de RAM por etiqueta de una salida del sistema)
 30 0000 080C SalFuzzy: EQU *
 31 * Velocidad del Rociador
 32 a00080C Parar RMB 1
 33 a00080D Lento RMB 1
 34 a00080E Medio RMB 1
 35 a00080F Rapido RMB 1
 36 a000810 Todo RMB 1
 37
 38 * salida crisp del Sistema (1 byte de RAM)
 39 a000811 VelRociador: RMB 1
 40
 41 ORG \$D00 ; inicia memoria con base de conocimiento
 42 *****
 43 * La siguiente es la base de conocimiento especifica de la aplicacion
 44 *****
 45 entrada_FPs: ; FP = Función de Pertenencia
 46 * Cada FP trapezoidal definida por 4 bytes -punto1,punto2,pendiente1,pendiente2
 47 * Temperatura
 48 a000D00 0058 0008 FCB \$00,\$58,\$00,\$08 ;FP para temp Helada
 49 a000D04 4088 0B0B FCB \$40,\$88,\$0B,\$0B ;FP para temp Fria
 50 a000D08 7898 2020 FCB \$78,\$98,\$20,\$20 ;FP para temp Normal
 51 a000D0C 88D0 0B0B FCB \$88,\$D0,\$0B,\$0B ;FP para temp Tibia
 52 a000D10 B8FF 0800 FCB \$B8,\$FF,\$08,\$00 ;FP para temp Caliente
 53 * Humedad
 54 a000D14 0040 0008 FCB \$00,\$40,\$00,\$08 ;FP para Humedad MuyBaja
 55 a000D18 2068 080B FCB \$20,\$68,\$08,\$0B ;FP para Humedad Baja
 56 a000D1C 5090 0B08 FCB \$50,\$90,\$0B,\$08 ;FP para Humedad Media

```

57 a000D20 70B0 080B FCB $70,$B0,$08,$0B ;FP para Humedad Alta
58 a000D24 98FF 0B00 FCB $98,$FF,$0B,$00 ;FP para Humedad MuyAlta
59
60 *****
61 * Fija sesgos de modo que las Reglas sean mas entendibles
62 * todos los sesgos relativos al inicio de EntFuzzy
63 *****
64 0000 0000 T_Helada EQU (Helada-EntFuzzy) ;sesgo para entradas fuzzy
65 0000 0001 T_Fria EQU (Fria-EntFuzzy)
66 0000 0002 T_Normal EQU (Normal-EntFuzzy)
67 0000 0003 T_Tibia EQU (Tibia-EntFuzzy)
68 0000 0004 T_Caliente EQU (Caliente-EntFuzzy)
69
70 0000 0005 H_MuyBaja EQU (MuyBaja-EntFuzzy)
71 0000 0006 H_Baja EQU (Baja-EntFuzzy)
72 0000 0007 H_Media EQU (Media-EntFuzzy)
73 0000 0008 H_Alta EQU (Alta-EntFuzzy)
74 0000 0009 H_MuyAlta EQU (MuyAlta-EntFuzzy)
75
76 0000 000A R_Parar EQU (Parar-EntFuzzy)
77 0000 000B R_Lento EQU (Lento-EntFuzzy)
78 0000 000C R_Medio EQU (Medio-EntFuzzy)
79 0000 000D R_Rapido EQU (Rapido-EntFuzzy)
80 0000 000E R_Todo EQU (Todo-EntFuzzy)
81
82 0000 00FE Sep EQU $FE ;Separador entre entradas/salidas
83 * El mismo $FE se usa para separar Reglas sucesivas
84 0000 00FF Regla_end EQU $FF ; terminador de Lista de Reglas
85
86 Regla_list:
87 * Este comentario muestra la primera Regla para ejemplificar.
88 * IF Temp es Helada AND Humedad es MuyBaja THEN poner Parar Rociador.
89 a000D28 0005 FE0A FCB T_Helada,H_MuyBaja,Sep,R_Parar,Sep ;Regla 1
90 000D2C FE
91 a000D2D 0006 FE0A FCB T_Helada,H_Baja,Sep,R_Parar,Sep ;Regla 2
92 000D31 FE
93 a000D32 0007 FE0A FCB T_Helada,H_Media,Sep,R_Parar,Sep ;Regla3

```

```

000D36 FE
92  a000D37 0008 FE0A FCB T_Helada,H_Alta,Sep,R_Parar,Sep ;Regla 4
    000D3B FE
93  a000D3C 0009 FE0A FCB T_Helada,H_MuyAlta,Sep,R_Parar,Sep ;Regla 5
    000D40 FE
94
95  a000D41 0105 FE0A FCB T_Fria,H_MuyBaja,Sep,R_Parar,Sep ;Regla 6
    000D45 FE
96  a000D46 0106 FE0A FCB T_Fria,H_Baja,Sep,R_Parar,Sep ;Regla 7
    000D4A FE
97  a000D4B 0107 FE0B FCB T_Fria,H_Media,Sep,R_Lento,Sep ;Regla 8
    000D4F FE
98  a000D50 0108 FE0B FCB T_Fria,H_Alta,Sep,R_Lento,Sep ;Regla 9
    000D54 FE
99  a000D55 0109 FE0B FCB T_Fria,H_MuyAlta,Sep,R_Lento,Sep ;Regla10
    000D59 FE
100
101 a000D5A 0205 FE0B FCB T_Normal,H_MuyBaja,Sep,R_Lento,Sep ;Regla
11
    000D5E FE
102 a000D5F 0206 FE0B FCB T_Normal,H_Baja,Sep,R_Lento,Sep ;Regla 12
    000D63 FE
103 a000D64 0207 FE0C FCB T_Normal,H_Media,Sep,R_Medio,Sep ;Regla 13
    000D68 FE
104 a000D69 0208 FE0C FCB T_Normal,H_Alta,Sep,R_Medio,Sep ;Regla 14
    000D6D FE
105 a000D6E 0209 FE0C FCB T_Normal,H_MuyAlta,Sep,R_Medio,Sep ;Regla
15
    000D72 FE
106
107 a000D73 0305 FE0B FCB T_Tibia,H_MuyBaja,Sep,R_Lento,Sep ;Regla 16
    000D77 FE
108 a000D78 0306 FE0C FCB T_Tibia,H_Baja,Sep,R_Medio,Sep ;Regla 17
    000D7C FE
109 a000D7D 0307 FE0D FCB T_Tibia,H_Media,Sep,R_Rapido,Sep ;Regla 18
    000D81 FE
110 a000D82 0308 FE0D FCB T_Tibia,H_Alta,Sep,R_Rapido,Sep ;Regla 19

```



```

000D86 FE
111 a000D87 0309 FE0D FCB T_Tibia,H_MuyAlta,Sep,R_Rapido,Sep ;Regla
20
000D8B FE
112
113 a000D8C 0405 FE0C FCB T_Caliente,H_MuyBaja,Sep,R_Medio,Sep
;Regla 21
000D90 FE
114 a000D91 0406 FE0D FCB T_Caliente,H_Baja,Sep,R_Rapido,Sep ;Regla
22
000D95 FE
115 a000D96 0407 FE0D FCB T_Caliente,H_Media,Sep,R_Rapido,Sep
;Regla 23
000D9A FE
116 a000D9B 0408 FE0E FCB T_Caliente,H_Alta,Sep,R_Todo,Sep ;Regla 24
000D9F FE
117 a000DA0 0409 FE0E FCB T_Caliente,H_MuyAlta,Sep,R_Todo ;Regla 25
118 a000DA4 FF FCB Regla_end
119
120 salida_FPs:
121 a000DA5 00 FCB $00 ;FP para Rociador Parar
122 a000DA6 30 FCB $30 ;FP para Rociador Lento
123 a000DA7 60 FCB $60 ;FP para Rociador Medio
124 a000DA8 A0 FCB $A0 ;FP para Rociador Rapido
125 a000DA9 FF FCB $FF ;FP para Rociador Todo
126 * Fin de la Base de Conocimiento
127
128 ORG $0E00 ; Empieza memoria del Programa
129 *****
130 * Lo siguiente es la parte ejecutable programa fuzzy
131 *****
132 a000E00 CE 0D00 Fuzzify: LDX #entrada_FPs ;Apunta a definiciones de FPs
133 a000E03 CD 0802 LDY #EntFuzzy ;Apunta a entradas fuzzy en RAM
134 a000E06 B6 0800 LDAA Temperatura ;Trae 1a entrada del sistema
135 a000E09 C605 LDAB #5 ;Temperatura tiene 5 etiquetas
136 a000E0B 01 Fuz_loop: MEM ;Evalua una FP de
137 a000E0C 0431 FC DBNE B,Fuz_loop ;para 5 etiquetas de 1er entrada

```

```

138  a000E0F B6 0801 LDAA Humedad ;Trae 2da entrada del sistema
139  a000E12 C605 LDAB #5 ;Humedad tiene 5 etiquetas
140  a000E14 01 Fuz_loop1: MEM ;Evalua una FP
141  a000E15 0431 FC DBNE B,Fuz_loop1 ;para las 5 etiquetas de 2da entrada
142  * Aqui X apunta a lista de Reglas, Y apunta a salidas fuzzy
143  a000E18 C605 LDAB #5 ;5 salidas fuzzy en RAM
144  a000E1A 6970 Regla_eval: CLR 1,Y+ ;Cero a salidas fuzzy e incr apuntr
145  a000E1C 0431 FB DBNE B,Regla_eval ;Lazo para limpiar todas las salidas fuzzy
146  a000E1F CD 0802 LDY #EntFuzzy ;Puntero a entradas y sal fuzzy
147  * X ya apunta al tope de la lista de Reglas
148  a000E22 86FF LDAA #$FF ;Inicia a A y limpia bit
V
149  a000E24 183A REV ;Procesa lista de Regla
150  * Aqui X apunta a FPs de salida, reg Y a entradas fuzzy
151  a000E26 CD 080C Defuz: LDY #SalFuzzy ; Apunta a salidas fuzzy
152  * X ya apunta a FPs tipo singleton
153  a000E29 C605 LDAB #5 ;5 salidas fuzzy por salida crisp
154  a000E2B 183C WAV ;Calcula sumas para prom. ponderado
155  a000E2D 11 EDIV ;Finalmente divide para prom ponderado
156  a000E2E B764 TFR Y,D ;Mueve resultado a A:B
157  a000E30 7B 0811 STAB VelRociador ;Salva salida crisp del sistema

```


A.3 ADQUISIDOR DE DATOS.

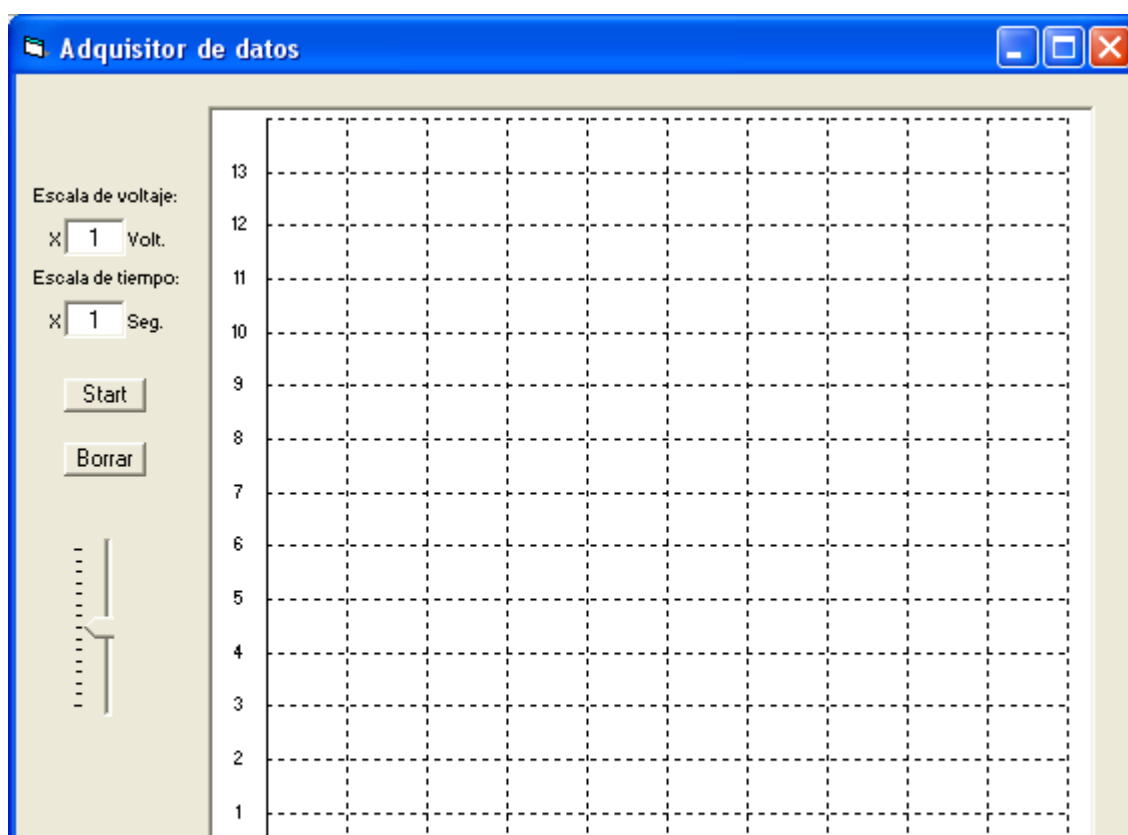
PRESENTACIÓN

Debido a la naturaleza del trabajo, se considero necesario el desarrollar una aplicación complementaria al simulador y controlador difuso, dicha aplicación es un adquisidor de datos hacia la PC para visualización de tendencias históricas.

La aplicación se desarrollo en Visual Basic y utiliza una subrutina de ensamblador para el microcontrolador Motorola 68HC12 para comunicarse vía SCI. Para la adquisición de la información se utiliza una subrutina interruptible.

Cuando se utiliza el Adquisidor de Datos para la recopilación de los datos se están guardando en un archivo el cual se puede manipular posteriormente.

Por medio del botón  se abre el adquisidor de datos se muestra la siguiente pantalla:



Los componentes de esta pantalla son: una cuadrícula, un seteo de la base de tiempo y un seteo de la escala de voltaje, un botón de Start y un botón de Stop y un botón de Cerrar.

La cuadrícula es la parte de la pantalla donde se grafica la señal proveniente del puerto PAD7 del Microcontrolador.

El botón de Start proporciona al usuario el control de cuando iniciar a graficar el resultado de la prueba que en ese momento se esta ejecutando.

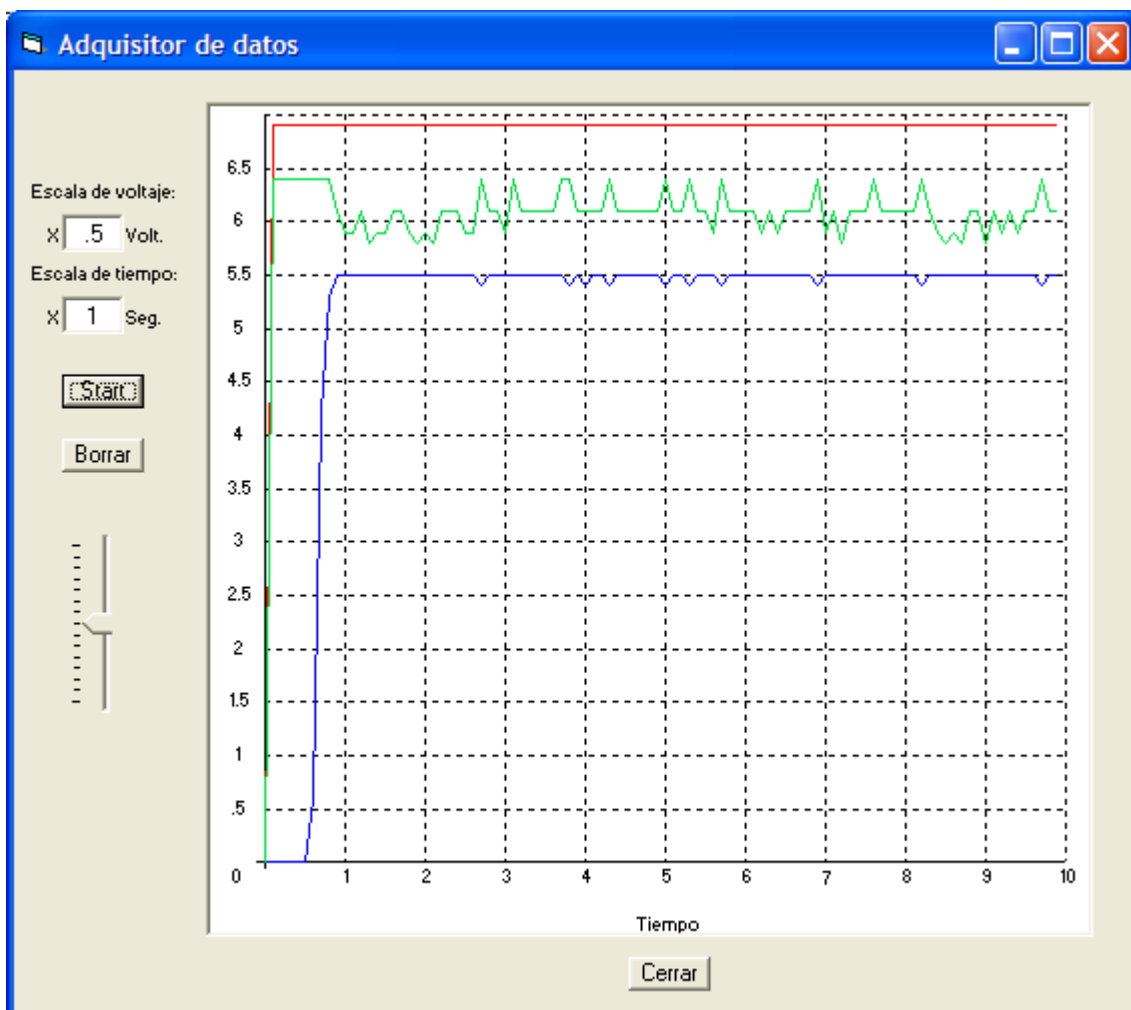
Con el botón de Stop se puede detener el proceso de graficar.

El cuadro de texto “Escala de Voltaje” permite ingresar una escala de voltaje deseada dependiendo de la magnitud del voltaje que se desee graficar.

El cuadro de texto Escala de Tiempo determina el tiempo de muestreo de la señal, el cual se puede variar según sea la respuesta del sistema que se tiene bajo prueba.

La base de tiempo para sistemas de control de temperatura es generalmente mayor que los sistemas como el control de velocidad en un motor y el sistema de control de nivel.

Ejemplo de traza obtenida con el Adquisidor de Datos:



CODIGO FUENTE DEL ADQUISIDOR DE DATOS

Option Explicit

Dim tiempo, referencia, lazo, step, X1(1 To 3), X2(1 To 3), Y1(1 To 3), Y2(1 To 3), Y(1 To 3) As Integer

Dim entrada As String

Dim escY, escX As Double

Dim base As Variant

Private Sub Command1_Click()

End

End Sub

Private Sub Command2_Click()

If Timer1.Enabled = False Then

 Picture1.Cls

 X1(1) = 400

 X2(1) = 400

 Y1(1) = 5660 - referencia * 400

 Y2(1) = 5660 - referencia * 400

 ' X1(2) = 400

 ' X2(2) = 400

 ' Y1(2) = 5660 - referencia * 400

 ' Y2(2) = 5660 - referencia * 400

 ' X1(3) = 400

 ' X2(3) = 400

 ' Y1(3) = 5660 - referencia * 400

 ' Y2(3) = 5660 - referencia * 400

 entrada = " "


```

step = 0

Y(1) = 0

Y(2) = 0

Y(3) = 0

If escX < 1 And escX > 0 Then

    Timer1.Interval = escX * 10

Else

    Timer1.Interval = escX * 100

End If

Timer1.Enabled = True

Timer2.Enabled = True

MSComm1.PortOpen = True

base = Time

Open "datos.txt" For Output As #1

tiempo = 0

Write #1, "Tiempo[x" + Str(Timer1.Interval) + "mSeg] Entrada[temp.]"

Command2.Caption = "Stop"

Label10.Caption = Format(Time - base, "hh:mm:ss")

Else

    Command2.Caption = "Start"

    MSComm1.PortOpen = False

    Timer1.Enabled = False

    Timer2.Enabled = False

    Close #1

End If

End Sub

Private Sub Command3_Click()

```

X1(1) = 400

X2(1) = 400

Y1(1) = 5660 - referencia * 400

Y2(1) = 5660 - referencia * 400

'X1(2) = 400

'X2(2) = 400

'Y1(2) = 5660 - referencia * 400

'Y2(2) = 5660 - referencia * 400

'X1(3) = 400

'X2(3) = 400

'Y1(3) = 5660 - referencia * 400

'Y2(3) = 5660 - referencia * 400

step = 0

Picture1.Cls

End Sub

Private Sub Form_Load()

MSComm1.InputMode = comInputModeText

MSComm1.InputLen = 1

escY = Val(Text1.Text)

escX = Val(Text2.Text)

End Sub

Private Sub Slider1_Change()

referencia = Slider1.Value

Call Text1_Change

End Sub

```
Private Sub Text1_Change()
```

```
    escY = Val(Text1.Text)
```

```
    For lazo = 0 To 12
```

```
        Label3(lazo).Caption = Str(((lazo + 1) + referencia) * escY)
```

```
    Next lazo
```

```
End Sub
```

```
Private Sub Text2_Change()
```

```
    escX = Val(Text2.Text)
```

```
    If escX >= 0.1 Then
```

```
        For lazo = 0 To 9
```

```
            Label4(lazo).Caption = Str((lazo + 1) * escX)
```

```
        Next lazo
```

```
    Else
```

```
        MsgBox "No se permite usar valores menores a 0.1"
```

```
        escX = 1
```

```
        Text2.Text = escX
```

```
    End If
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    tiempo = tiempo + 1
```

```
    step = step + 1
```

```
    MSComm1.Output = "1"
```

```
    Do
```

```
        Loop Until MSComm1.OutBufferCount = 0
```

```
    Do
```

```
        Loop Until MSComm1.InBufferCount = 1
```

```

entrada = MSComm1.Input

Y(1) = 0.628272251 * (Asc(entrada) - 47) + 25.1

'Do

'Loop Until MSComm1.InBufferCount = 1

'entrada = MSComm1.Input

'Y(2) = Asc(entrada) / 3.44

'Do

'Loop Until MSComm1.InBufferCount = 1

'entrada = MSComm1.Input

'Y(3) = Asc(entrada) / 2

'If Y(1) > 127 Then Y(1) = Y(1) - 255

X2(1) = 400 + step * 60

Y2(1) = 5660 - (Y(1) * 40 / escY) * 10 + (referencia * 400)

Write #1, tiempo, " ", Y(1)

'If Y(2) > 127 Then Y(2) = Y(2) - 255

'X2(2) = 400 + step * 60

'Y2(2) = 5660 - (Y(2) * 40 / escY) + (referencia * 400)

'If Y(3) > 127 Then Y(3) = Y(3) - 255

'X2(3) = 400 + step * 60

'Y2(3) = 5660 - (Y(3) * 40 / escY) + (referencia * 400)


If step * 60 <= 6000 Then

    Picture1.Line (X1(1), Y1(1))-(X2(1), Y2(1)), &HFF

'    Picture1.Line (X1(2), Y1(2))-(X2(2), Y2(2)), &HFF0000

'    Picture1.Line (X1(3), Y1(3))-(X2(3), Y2(3)), &H40E000

'    For lazo = 1 To 3

        X1(1) = X2(1)

        Y1(1) = Y2(1)

```

```
' Next lazo
```

```
Else
```

```
For lazo = 1 To 3
```

```
    X1(1) = 400
```

```
    X2(1) = 400
```

```
Next lazo
```

```
step = 0
```

```
Picture1.Cls
```

```
End If
```

```
End Sub
```

```
Private Sub Timer2_Timer()
```

```
Label10.Caption = Format(Time - base, "hh:mm:ss")
```

```
End Sub
```

A.4 ACONDICIONADOR DE SEÑAL

Debido a las características eléctricas inherentes de la tarjeta de evaluación se requirió de la implementación de circuitos acondicionadores de señal tanto para los valores de entrada como los de salida, los cuales están limitados de 0 a 5 Voltios en ambas direcciones.

Las magnitudes hacia las cuales debe acondicionarse dependen en el caso de las entradas de los transductores seleccionados, ya sean señales de corriente, voltaje o resistencia; para el caso de las salidas dependen de las características eléctricas y/o mecánicas del actuador. Por ejemplo en la aplicación de control difuso sobre la centrifugadora de aceite el actuador es un conjunto de válvula, motor y pistón que actúa como un restrictor del caudal de vapor hacia la cámara intercambiadora de calor.

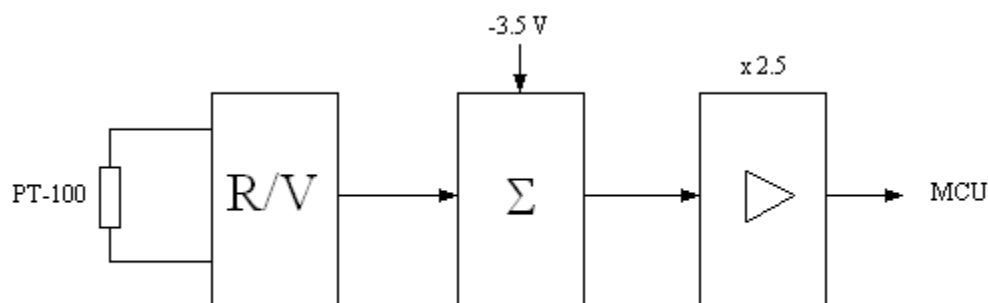


Figura 1. Acondicionador de señal para controlador de temperatura en centrifugadora.

En la figura 1 se presenta el diagrama de bloques del acondicionador de señal para el controlador de temperatura en la centrifugadora de aceite. En este caso el sensor de temperatura es un PT-100, el cual varía su resistencia cuando varía la temperatura, por lo que se implementó un circuito que proporcione un voltaje proporcional a dicha variación de temperatura, posteriormente y debido a la curva que se obtuvo con el PT-100 se requirió

corregir un offset de 3.5 voltios y luego aplicar una ganancia de 2.5 para obtener una señal de 0 a 5Vdc que es suministrada a la entrada del convertidor del MCU.