

**PROTOTIPO DE UNA HERRAMIENTA COMPUTACIONAL PARA
IMPLEMENTAR MODELOS FUZZY DE DOS CAPAS Y PERMITIR ANÁLISIS DE
SENSIBILIDAD DE MODELOS REPRESENTADOS USANDO LÓGICA FUZZY**

**PAOLA ANDREA LÓPEZ CORREA
JOHANNA MARITZA PATIÑO TEJEDOR**

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

2009

**PROTOTIPO DE UNA HERRAMIENTA COMPUTACIONAL PARA
IMPLEMENTAR MODELOS FUZZY DE DOS CAPAS Y PERMITIR ANÁLISIS DE
SENSIBILIDAD DE MODELOS REPRESENTADOS USANDO LÓGICA FUZZY**

**PAOLA ANDREA LÓPEZ CORREA
JOHANNA MARITZA PATIÑO TEJEDOR**

Proyecto de grado presentado como requisito para optar al título de Ingeniero de
Sistemas

Director

FERNANDO RUÍZ DÍAZ

Profesor Titular

Escuela de Ingeniería de Sistemas – UIS

Codirector

FREDY ALEXANDER ESTEPA MARTÍNEZ

Ingeniero de Sistemas – UIS

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

2009

DEDICATORIA

A Dios, gracias por tu profundo amor, sabiduría y por llenarme de la fuerza, la perseverancia y la paciencia necesarias para alcanzar ésta meta tan anhelada.

A mi Janecilla, gracias mami por tu apoyo incondicional, tus palabras de aliento en los momentos más difíciles, tu paciencia y por enseñarme a no darme por vencida y a seguir adelante a pesar de todo, TE ADORO MAMI, gracias por hacer de mí la mujer que soy ahora. A mis hermanitos Magdis y Migue por estar pendientes de mí y por darme ánimo siempre que lo necesité, LOS ADORO.

A mi tía Cecy, gracias por tus valiosos consejos, por estar pendiente y cuidar de mí todo el tiempo, por tus oraciones y tu cariño que me dieron valor y fuerza para continuar en los momentos más duros, gracias tía, TE QUIERO MUCHO, a mis primos Danna y Carlos Andrés porque junto con mi tía han sido como una segunda familia para mí, los quiero muchísimo.

Por último, pero no menos importante, a mi Johis por tu valiosa amistad, gracias amiguita por tu compañía, eres una personita muy especial, siempre ocuparás un lugar en mi corazón, TE QUIERO AMIGUITA.

Paola Andrea C.

DEDICATORIA

A Dios, por estar con migo y darme la fortaleza, sabiduría y perseverancia para llevar este proyecto por el mejor camino y alcanzar este logro tan importante.

A mi mamita Dora por cada una de las palabras de aliento que tuvo para hacerme seguir adelante, por brindarme su apoyo incondicional, sus enseñanzas y sobretodo su amor, por la confianza que ha depositado en mí y por todas las cosas lindas que ha puesto en mi vida.

A mi tío Gerardo, por creer en mí y darme la oportunidad de cumplir este sueño, por sus consejos, opiniones y sugerencias que de una u otra manera influyen en mi formación personal y profesional.

A mi familia en general, por acompañarme, apoyarme y demostrarme la importancia de tenerlos a mi lado.

A mi amiguita Paola por su amistad, tolerancia y honestidad, por estar con migo en los momentos difíciles y por enseñarme a mejorar cada día.

Por último, agradezco a cada uno de mis amigos, por haberlos conocido, por compartir este triunfo y por hacer que los lleve en el corazón, muchas gracias a todos ustedes, los quiero mucho.

Johanna Patiño Tejedor

AGRADECIMIENTOS

A la Universidad Industrial De Santander UIS y la Escuela de Ingeniería de Sistemas e Informática por habernos brindado los conocimientos necesarios que nos consolidan hoy como unas excelentes profesionales.

A nuestro director de proyecto Fernando Ruíz, gracias profe por confiar en nosotras y por dedicarnos parte de su tiempo para sacar éste proyecto adelante, además por sus palabras de reconocimiento, MUCHAS GRACIAS PROFESOR.

Al Ingeniero Fredy Estepa Martínez por colaborarnos en el desarrollo de este proyecto, por supervisarnos y hacer posible que todo saliera bien, por sus aportes y por enseñarnos nuevos conocimientos que estamos seguras influirá en gran medida nuestra formación profesional.

TABLA DE CONTENIDO

CAPÍTULO I	1
1. DESCRIPCIÓN DEL PROYECTO	1
1.1. OBJETIVOS DEL PROYECTO	1
1.1.1. OBJETIVO GENERAL.....	1
1.1.2. OBJETIVOS ESPECÍFICOS	1
1.2. JUSTIFICACIÓN.....	3
1.2.1. DEFINICIÓN DEL PROBLEMA	3
1.3. IMPACTO.....	4
1.4. VIABILIDAD.....	5
CAPÍTULO II	6
2. MARCO TEÓRICO	6
2.1. LÓGICA DIFUSA	6
2.1.1. INTRODUCCIÓN.....	6
2.1.2. RESEÑA HISTÓRICA	7
2.1.3. GENERALIDADES.....	8
2.1.4. COMPARACIÓN ENTRE LÓGICA CLÁSICA Y LÓGICA DIFUSA.....	11
2.1.5. FUNCIONES DE PERTENENCIA.....	13
2.1.6. OPERACIONES BÁSICAS CON CONJUNTOS BORROSOS.....	17
2.1.7. VARIABLES LINGÜÍSTICAS.....	20
2.1.8. SISTEMAS BASADOS EN LOGICA DIFUSA.....	21
2.1.9. SISTEMAS DIFUSOS DE DOS CAPAS O NIVELES	27
2.2. APLICACIONES DE LOS SISTEMAS DIFUSOS DE DOS CAPAS.....	28
2.2.1. PLANIFICACIÓN DEL DISEÑO DE PLANTAS.....	28
2.2.2. DETERMINACIÓN DE LA PROBABILIDAD DE PICOS TRANSVALVULARES EN ECOCARDIOGRAMAS	39
2.3. METODOLOGÍA DE DESARROLLO APLICADA.....	49

2.3.1. PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE	50
2.3.2. DIMENSIÓN TEMPORAL Y DIMENSIÓN ESTÁTICA.....	51
2.3.3. LENGUAJE DE MODELADO UNIFICADO	57
2.3.4. DIAGRAMAS UML	58
2.4. TECNOLOGÍA DE DESARROLLO UTILIZADA.....	61
2.4.1. JAVA.....	61
2.4.2. ARCHIVOS XML	63
2.4.3. ANÁLISIS DE SENSIBILIDAD DE MODELOS.....	66
2.4.4. PATRÓN DE DESARROLLO DE SOFTWARE.....	67
CAPÍTULO III.....	69
3. MARCO METODOLÓGICO.....	69
3.1. FASE DE INICIO.....	69
3.1.1. PLANEACIÓN DE LA FASE:.....	69
3.1.2. ANÁLISIS DEL ALCANCE	70
3.1.3. DESCRIPCIÓN DE NECESIDADES:	72
3.1.4. FUNCIONES DEL PRODUCTO	74
3.1.5. RIESGOS DEL PROYECTO:.....	75
3.2. FASE DE ELABORACIÓN.....	76
3.2.1. PLANEACIÓN DE LA FASE:.....	76
3.2.2. DEFINICIÓN DE REQUISITOS	77
3.2.3. ARQUITECTURA SOFTWARE	81
3.3. FASE DE CONTRUCCIÓN.....	110
3.3.1. PLANEACIÓN DE LA FASE:.....	110
3.3.2. SOFTWARE Y LENGUAJE DE PROGRAMACIÓN UTILIZADO	111
3.3.3. DESCRIPCIÓN DE LOS COMPONENTES DE LA APLICACIÓN	111
3.4. FASE DE TRANSICIÓN	127
3.4.1. ALCANCE DE LA FASE.....	127
3.4.2. ESPECIFICACIÓN DE PRUEBAS	127
3.4.3. MANUAL DE USUARIO	128

CAPÍTULO IV	145
4. PRUEBAS Y RESULTADOS	145
4.1. <i>COMPARACIÓN DE RESULTADOS ENTRE EL PROTOTIPO Y MATLAB</i>	145
4.2. <i>APLICACIÓN DE LOS MODELOS DE DISEÑO DE PLANTAS Y DETERMINACIÓN DE PICOS EN EL FLUJO TRANS-MITRAL Y TRANS-AÓRTICO DEL CORAZÓN</i> ...	148
4.2.1. DISEÑO DE PLANTAS:	148
4.2.2. DETERMINACIÓN DE PICOS EN EL FLUJO TRANS-MITRAL Y TRANS-AÓRTICO DEL CORAZÓN	150
4.3. ESTIMACIÓN DE LOS TIEMPOS DE PROCESAMIENTO DEL PROTOTIPO	153
CAPÍTULO V	155
5. CONCLUSIONES	155
CAPÍTULO VI	157
6. RECOMENDACIONES Y TRABAJOS FUTUROS	157
CAPÍTULO VII	158
7. BIBLIOGRAFÍA	158

LISTA DE FIGURAS

Figura 1. Diagrama conceptual entre la lógica clásica y la lógica difusa	11
Figura 2. Lógica Clásica Vs. Lógica Difusa.....	12
Figura 3. Función de pertenencia Triangular	14
Figura 4. Función de pertenencia Trapezoidal.....	15
Figura 5. Función de pertenencia Gaussiana	16
Figura 6. Función de pertenencia Campana de Bell.....	16
Figura 7. Función de pertenencia Sigmoidal.....	17
Figura 8. Ejemplos de conjuntos borrosos.....	18
Figura 9. Unión de dos conjuntos borrosos	19
Figura 10. Intersección de dos conjuntos borrosos.....	19
Figura 11. Complemento de un conjunto difuso.....	20
Figura 12. Esquema de los componentes de una variable lingüística.	21
Figura 13. Esquema General de un Sistema Basado en Lógica Difusa.	22
Figura 14. Sistema Difuso Mamdani. Proceso general.....	23
Figura 15. Partes de una regla difusa de tipo Mamdani.....	24
Figura 16. Sistema Difuso Sugeno. Proceso General.....	25
Figura 17. Partes de una regla difusa de tipo Sugeno.	26
Figura 18. Estructura General de un Modelo Difuso de dos capas.....	28
Figura 19. Variables de Entrada Modelo Diseño de Plantas (Capa 1).....	34
Figura 20. Variables de Salida Capa 1 y Entrada Capa 2 del Modelo Diseño de Plantas.....	36
Figura 21. Índice de Cercanía.....	37
Figura 22. Modelo Difuso de dos capas para la planificación de plantas.....	38
Figura 23. Esquema General del Sistema de Detección de Tiempos de Inicio y Fin de Ondas de Flujo Trans-valvular.....	42

Figura 24. Variables de Entrada Modelo Probabilidad de Picos en Flujo Trans-Mitral y Trans-Aortico.....	46
Figura 25. Variables de la Capa 2 Modelo Probabilidad de Picos en Flujo Trans-Mitral y Trans-Aortico.....	47
Figura 26. Variables de Salida de la capa 2, PP_AberturaE, PP_CierreE, PP_AberturaBpc, PP_CierreBpc.....	47
Figura 27. Esquema General Modelo Difuso.....	48
Figura 28. Fases, Iteraciones y Disciplinas de la Metodología RUP.....	52
Figura 29. Tiempo de Ejecución de las Disciplinas en cada Fase.....	53
Figura 30. Diagrama Caso de Uso Usuario.....	82
Figura 31. Diagrama Caso de Uso Examinar Modelo.....	85
Figura 32. Diagrama Caso de Uso Gestionar Variable Lingüística.....	87
Figura 33. Diagrama Caso de Uso Gestionar Reglas de Inferencia.....	90
Figura 34. Diagrama Caso de Uso Visualizar Resultados.....	91
Figura 35. Diagrama Caso de Uso Realizar Análisis de Sensibilidad.....	93
Figura 36. Diagrama de Actividades Gestionar Variables Lingüísticas.....	95
Figura 37. Diagrama de Actividades Gestionar Reglas.....	96
Figura 38. Diagrama de Actividades Visualizar Resultados.....	97
Figura 39. Diagrama de Actividades Análisis de Sensibilidad.....	98
Figura 40. Diagrama de clases Paquete Datos.....	1
Figura 41. Diagrama de Clases Paquete Lógica.....	1
Figura 42. Diagrama de clases Paquete Función Pertenencia.....	1
Figura 43. Ventana esquema.....	103
Figura 44. Ventana Variable Lingüística.....	104
Figura 45. Ventana Reglas Capa 1.....	105
Figura 46. Ventana Reglas Capa 2.....	106
Figura 47. Ventana Resultados Capa 1.....	107
Figura 48. Ventana Resultados Capa 2.....	108
Figura 49. Ventana Estabilidad.....	109

Figura 50. Ventana Máximos y Mínimos.....	110
Figura 51. Árbol de Directorio de presentación en el paquete de Información Presentación.....	112
Figura 52. Árbol de Directorio del Modelo Vista Controlador MVC en el Paquete de Información Lógica.....	114
Figura 53. Árbol de directorio de la variable Lingüística, Conjunto Borrosos y Variables de Entrada y Salida en el Paquete de Información Lógica.....	115
Figura 54. Árbol de directorio de la variable Lingüística y Conjunto Borroso en el Paquete de Información Datos.....	116
Figura 55. Árbol de directorio de la regla de inferencia en los Paquetes de Información Datos y Lógica.	117
Figura 56. Árbol de Directorio de Funciones de Pertenencia en los Paquetes de información Datos y Lógica.....	119
Figura 57. Árbol de directorio de la Manipulación de la Persistencia de Datos en el Paquete de Información Datos.....	123
Figura 58. Serializacion Funciones de Pertenencia, Conjuntos Borrosos y Variables Lingüísticas.....	124
Figura 59. Serialización Reglas de Inferencia.....	125
Figura 60. Serialización Modelo.....	126
Figura 61. Distribución de Menús.	128
Figura 62. Ventana Esquema.	129
Figura 63. Ventana Variable Lingüística.	130
Figura 64. Ventana Reglas Capa 1 / Capa 2.	131
Figura 65. Ventana Resultados Capa1.....	131
Figura 66. Ventana Resultados Capa2.....	132
Figura 67. Ventana Estabilidad.....	132
Figura 68. Ventana Máximos y Mínimos.....	133
Figura 69. Menú Archivo.....	134
Figura 70. Menú Modelo.....	135

Figura 71. Menú Análisis de Sensibilidad.	136
Figura 72. Ventana Esquema.	137
Figura 73. Ventana Variable Lingüística.	138
Figura 74. Ventana Reglas Capa 1.	139
Figura 75. Ventana Reglas Capa 2.	140
Figura 76. Ventana Resultados Capa 1.	141
Figura 77. Ventana Resultados Capa 2.	142
Figura 78. Ventana Estabilidad.	143
Figura 79. Ventana Máximos y Mínimos.	144

LISTA DE TABLAS

Tabla 1. Descripción Funcional de los Involucrados en el Proyecto.....	57
Tabla 2. Descripción de las Principales Necesidades del Proyecto.	74
Tabla 3. Riesgos del Proyecto.	76
Tabla 4. Requisitos de la Herramienta.	80
Tabla 5. Descripción Caso de Uso Usuario.	84
Tabla 6. Descripción Caso de Uso examinar Modelo.	86
Tabla 7. Descripción Caso de Uso Gestionar Variable Lingüística.....	89
Tabla 8. Descripción Caso de Uso Gestionar Regla de Inferencia.....	91
Tabla 9. Descripción Caso de Uso Visualizar Resultados.	92
Tabla 10. Descripción Caso de Uso realizar Análisis de sensibilidad.	94
Tabla 11. Datos Utilizados en la Comparación de Resultados.....	147
Tabla 12. Resultados del Porcentaje Error Relativo.....	147
Tabla 13. Valores de las variables del Modelo Diseño de Plantas.	150
Tabla 14. Valores de las variables del Modelo Determinación de Picos de los Flujos Trans-Mitral.	151
Tabla 15. Valores de las variables del Modelo Determinación de Picos de los Flujos Trans-Aórticos.	152
Tabla 16. Tiempos de Procesamiento de la Ventana Estabilidad.....	154
Tabla 17. Tiempos de Procesamiento de la Ventana Máximos y Mínimos.	154

RESUMEN

Título:

PROTOTIPO DE UNA HERRAMIENTA COMPUTACIONAL PARA IMPLEMENTAR MODELOS FUZZY DE DOS CAPAS Y PERMITIR ANÁLISIS DE SENSIBILIDAD DE MODELOS REPRESENTADOS USANDO LÓGICA FUZZY*

Autores:

JOHANNA MARITZA PATIÑO TEJEDOR**

PAOLA ANDREA LÓPEZ CORREA**

Palabras Claves: Lógica Fuzzy, dos capas, variables lingüísticas, funciones de pertenencia, conjuntos difusos, modelos difusos

Descripción

El prototipo de herramienta computacional FusysTwo versión 1.0 permite la creación y elaboración de modelos difusos de dos capas utilizando lógica fuzzy, permitiendo realizar análisis de sensibilidad de modelos representados de tal manera que se muestre el comportamiento de la estabilidad y la susceptibilidad de resultados ante cambios en las condiciones iniciales. Los sistemas difusos de dos capas facilitan el modelamiento de situaciones específicas y de mayor complejidad, convirtiéndose en un apoyo para los procesos de toma de decisiones y solución de problemas que al integrarse con el análisis de sensibilidad brindan la posibilidad de estimar la eficiencia de un modelo en base a las condiciones que lo afectan. La lógica difusa provee soporte al traducir el conocimiento heurístico experimentado de frases lingüísticas imprecisas a algoritmos numéricos, fundamentada en la teoría de conjuntos, permite imitar el comportamiento de la lógica humana y siendo un tipo de lógica flexible expresa un razonamiento aproximado aplicable a conceptos que toman valores de veracidad establecidos dentro de un rango de valores de 0 a 1, haciendo uso de variables lingüísticas conformadas por conjuntos difusos y funciones de pertenencia. FusysTwo contribuye a los estudiantes como herramienta de trabajo en materias relacionadas con lógica difusa para lograr que los usuarios extiendan sus conocimientos y reconozcan la importancia de utilizarla, además de ser útil en áreas como la ingeniería, los procesos de manufactura y la medicina entre otros. El manual de usuario de FusysTwo representa esquemática y gráficamente su funcionamiento permitiendo guiar al usuario fácilmente.

* Trabajo de Grado

** Facultad de ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Fernando Ruíz Díaz. Codirector Fredy Estepa Martínez.

ABSTRACT

Title:

PROTOTYPE OF A COMPUTATIONAL TOOL FOR IMPLEMENTING TWO-LAYER FUZZY MODELS AND ALLOW SENSITIVITY ANALYSIS FOR REPRESENTED MODELS USING FUZZY LOGIC*

Authors:

JOHANNA MARITZA PATIÑO TEJEDOR**

PAOLA ANDREA LOPEZ CORREA**

Keywords: Fuzzy logic, two layers, linguistic variables, membership functions, fuzzy sets, fuzzy model

Description

The prototype computer tool FusysTwo version 1.0 enables the creation and development of two-layer fuzzy models using fuzzy logic, allowing sensitivity analysis of represented models to show the behavior of the stability and sensitivity of results to changes in the initial conditions. The two-layer fuzzy systems facilitate modeling specific and complex situations, becoming a support for decision making and problem-solving that integrated with sensitivity analysis offer the possibility to estimate the efficiency of a model based on the conditions affecting it. The fuzzy logic provides support to translate the experienced heuristic knowledge from vague language phrases to numerical algorithms, based on set theory, allows to mimic the behavior of human logic and being a type of flexible logic states an approximate reasoning that applies to concepts that take values of accuracy within a set range of values between 0 to 1, making use of linguistic variables including fuzzy sets and membership functions. FusysTwo contributes to students as a working tool in areas related to fuzzy logic allowing users to extend their knowledge and recognizing the importance of using it, in addition of being useful in areas such as engineering, manufacturing processes and medicine, among others. The user manual of FusysTwo represents schematic and graphically its operation allowing the user to be easily guided.

* Graduate Thesis Project

** Physical Mechanical Engineering Faculty. School of IT Engineering and Informatics.

Director Fernando Ruíz Díaz. Codirector Fredy Estepa Martínez.

CAPÍTULO I

1. DESCRIPCIÓN DEL PROYECTO

1.1. OBJETIVOS DEL PROYECTO

1.1.1. OBJETIVO GENERAL

Desarrollar un prototipo de una herramienta computacional que permita representar modelos FUZZY de dos capas y a través de la cual se pueda realizar un análisis de sensibilidad de los modelos representados.

1.1.2. OBJETIVOS ESPECÍFICOS

1.1.2.1. Diseñar e implementar un prototipo de herramienta computacional basada en lógica FUZZY¹ que permita:

Representar modelos FUZZY de dos capas, demostrando su utilidad en problemas extraídos de artículos publicados, realizando una comprobación de su eficacia a través una comparación de resultados con Matlab.

Visualizar el proceso de solución aproximada del sistema a través del modelo para brindar comprensión al usuario.

¹ Lógica FUZZY también llamada lógica borrosa o difusa

Realizar un análisis de sensibilidad de los modelos de tal manera que se permita mostrar al usuario el comportamiento de la estabilidad de los modelos representados y la susceptibilidad de los resultados al cambio de las condiciones.

1.1.2.2. Realizar un manual de usuario con una descripción esquemática y gráfica del prototipo de la herramienta computacional que facilite su uso.

1.2. JUSTIFICACIÓN

1.2.1. DEFINICIÓN DEL PROBLEMA

La llegada de la lógica difusa a campos como el del control automático de procesos ha brindado a los expertos una gran alternativa a la hora de generar resultados lo suficientemente satisfactorios para resolver problemas de elevada complejidad y fuertemente relacionados con la realidad, cosa que hace algún tiempo era sumamente complicada.

De manera especial se hace énfasis en que el éxito de dicha tecnología se debe a su estrecha relación con el comportamiento de la lógica del ser humano centrada en el uso de etiquetas lingüísticas, estableciendo así una base importante dentro de la inteligencia artificial, hecho gracias al cual se obtienen soluciones cercanas a la realidad.

Actualmente existen herramientas que permiten encontrar soluciones a problemas complejos, haciendo uso de la lógica difusa, resaltando el hecho de que dichas herramientas solo permiten crear modelos de una capa, razón por la cual se dificulta o imposibilita la representación de algunos problemas donde es necesario hacer uso de modelos de dos capas los cuales en la mayoría de los casos, facilitan la representación de los problemas y se acercan a la realidad; además, las herramientas actuales tampoco cuentan con recursos que permitan realizar análisis de sensibilidad de los modelos diseñados y no ofrecen opciones para evaluar la estabilidad del modelo, esto sería útil para identificar como el cambio en una variable afecta el resultado de todo el modelo, además aquellas variables que son críticas y los diferentes escenarios que permiten analizar el comportamiento de un resultado ante diferentes valores en las entradas, estos análisis permiten entender mejor los modelos y facilitan la interpretación de sus resultados; por tal

motivo con éste proyecto se pretende dar una alternativa de solución a estos inconvenientes.

1.3. IMPACTO

Primero se resalta que el presente proyecto crea cierto impacto en distintos niveles como el social, el económico y el técnico; socialmente se encuentra que el software ofrece una interfaz fácil de usar por cualquier tipo de usuario lo que conlleva a facilitar el trabajo interdisciplinario entre distintos profesionales involucrados en proyectos relacionados con cualquier rama de investigación, es decir, el software facilita el trabajo en equipo y es aplicable a distintos escenarios. Económicamente, reduce los costos vinculados a la búsqueda de soluciones óptimas al disminuir el tiempo de respuesta a un problema y al proveer resultados aproximados sobre lo que se espera, ya que existen situaciones que requieren un procedimiento extenso y que por medio de esta herramienta se llegan a disminuir los tiempos de ejecución y se aminoran los errores. A nivel técnico, con este proyecto se pretende establecer un prototipo de herramienta computacional que permita representar modelos en los que se tienen en cuenta aplicaciones de dos capas los cuales brindan un mejor entendimiento a quienes la utilicen; además cuenta con un manual que guía al usuario en la manipulación de la misma, esta herramienta realiza un análisis de sensibilidad que interpreta el comportamiento que tienen los modelos a partir de cambios en los valores de las variables, y la susceptibilidad de los resultados debido a variaciones en las condiciones, permitiendo con esto revalidar los modelos existentes o ajustarlos de acuerdo al conocimiento de los expertos.

1.4. VIABILIDAD

El proyecto es viable pues se cuenta con los recursos técnicos y de personal para realizarlo; a nivel de recurso humano se cuenta con la experiencia, y los conocimientos del profesor Fernando Ruíz, y el ingeniero de sistemas Fredy Estepa. Además de las condiciones mencionadas anteriormente se cuenta con la disponibilidad de tiempo de las realizadoras y de los equipos necesarios para llevar a cabo el proyecto.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. LÓGICA DIFUSA

2.1.1. INTRODUCCIÓN

La lógica difusa es una de las tecnologías con mayor crecimiento a nivel mundial por su simplicidad y amplio rango de aplicaciones, ya que se aplica a conceptos que toman un valor cualquiera de veracidad dentro de un conjunto de valores que oscilan entre dos extremos, uno absolutamente verdadero y otro definitivamente falso; es aplicable para trabajar con información imprecisa, por ejemplo, temperatura alta, densidad media, etc.; en términos de conjuntos borrosos que se combinan en reglas para dar lugar a resultados, por ejemplo, si la presión es alta reducir la temperatura. Los sistemas de control basados en lógica difusa combinan variables de entrada, definidas con conjuntos difusos, a través de reglas que producen uno o más valores de salida. Una de las mayores ventajas de los sistemas basados en lógica difusa es su facilidad de manejo de la incertidumbre y la vaguedad de los datos.

Dado que los sistemas que usan lógica difusa funcionan según reglas, tienen múltiples entradas y salidas pero es importante resaltar que si al combinarlas en reglas para obtener resultados, existen demasiadas entradas, esto dará lugar a un sistema complejo por eso es preferible, en éste caso, que existan subsistemas

cada uno con funciones limitadas, dando lugar a la presencia de modelos difusos de múltiples capas².

2.1.2. RESEÑA HISTÓRICA

Aunque la lógica difusa es conocida así desde 1965 cuando Lofty A. Zadeh la bautizó de ésta manera, se le dio este nombre gracias a los conjuntos borrosos (Fuzzy Sets); fue investigada por primera vez en la Universidad de Berkeley (California) por el ingeniero iraní Lofty A. Zadeh nacionalizado en USA, profesor en las más prestigiosas universidades norteamericanas y doctor honoris causa de varias instituciones académicas. Muchas de las obras de otros pensadores en distintas disciplinas que tenían una visión similar alejada de la lógica tradicional, influyeron en las tesis de Zadeh quién posteriormente publicó su ensayo “*Fuzzy sets*” e “*Informations and Control*”, ambos en 1965 y más tarde “Fuzzy algorithm” en 1968.

La inquietud de Zadeh por éste tema comenzó cuando se dio cuenta de lo que él llamo principio de incompatibilidad: “*Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes*”³, fue entonces cuando introdujo la idea de los conjuntos difusos los cuales representan de manera cercana el lenguaje natural del pensamiento humano, involucrando vaguedad y términos imprecisos,

² Para el presente documento solo se tratarán los modelos difusos de dos capas.

³ Tomado del artículo en internet llamado “*Descripción General de las Técnicas de Lógica Difusa*” que se puede encontrar en: http://www.tdr.cesca.es/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf

pues no usa números, como en la lógica clásica, sino etiquetas lingüísticas que tienden a expresar ideas subjetivas.

Una de las primeras aplicaciones de la lógica fuzzy se llevo a cabo en 1974 en Inglaterra por Mamdani y Assilian quienes desarrollaron el primer controlador difuso diseñado para una máquina de vapor, pero la primera implantación real de un controlador de éstos fue realizada en 1980 por F.L. Smith & Co. en una planta cementera en Dinamarca; al mismo tiempo otros de los investigadores continuaron indagando y haciendo aportes a la teoría de la lógica fuzzy, es así como Takagi y Sugeno desarrollan la primera aproximación para construir reglas difusas; luego aparecieron las redes neuronales y debido a su vínculo con los sistemas fuzzy y al interés que despertaron entre los científicos aparecieron los llamados *neuro-fuzzy systems*⁴, en los años noventa también aparecen los algoritmos genéticos; éstas 3 técnicas computacionales combinadas dan origen a herramientas de trabajo efectivas en el campo de los sistemas de control ya que las tres son complementarias y útiles en ésta área. Gracias a los sistemas de control difuso han aparecido diversos inventos tecnológicos como los sensores para la estabilización de las imágenes en las cámaras fotográficas y de video, las lavadoras, los controles para las maniobras de los aviones, las ollas arroceras, sistemas de frenos y el aire acondicionado, entre otros.

2.1.3. GENERALIDADES

Dado que el pensamiento humano común se expresa a través de etiquetas lingüísticas o de forma cualitativa y no cuantitativa; la lógica difusa permite representarlo mediante un lenguaje matemático basado en la teoría de conjuntos

⁴ Sistemas neuro-difusos son sistemas fuzzy que usan métodos de aprendizaje basados en redes neuronales para identificar y optimizar sus parámetros.

difusos y las funciones de pertenencia asociadas a ellos; los términos lingüísticos, son menos exactos pero proporcionan información relevante y comprensible para el razonamiento humano.

La restricción del cálculo proposicional clásico a sólo dos valores lógicos ha dado lugar a problemas no posibles de solucionar debido a la naturaleza bivaluada de las proposiciones, éste es el principal inconveniente al que la lógica difusa da solución, gracias a que es multivaluada adopta valores de verdad intermedios que establecen una transición gradual sin fronteras definidas, por eso el lenguaje natural es difuso pues involucra vaguedad y términos imprecisos.

Conceptos como la estatura de una persona son proposiciones difusas cuando se dice un hombre es alto, mediano o bajo. La lógica difusa entonces ofrece flexibilidad, tolerancia a la imprecisión, capacidad para modelar problemas no-lineales con base en el lenguaje natural.

En general, la lógica difusa se aplica tanto a sistemas de control como a sistemas de ingeniería, economía, física, biología, etc.; Se fundamenta en variables lingüísticas de entrada y salida del modelo, compuestas por conjuntos difusos y donde un valor perteneciente a ellos se evalúa según reglas de la forma “SI...ENTONCES...”, dando lugar a otro valor de salida.

El valor de verdad asignado a una proposición fuzzy puede ser cualquiera dentro de un intervalo $[0,1]$. Esto se puede expresar como:

$$T: x \in X \rightarrow \{0,1\} \quad (1)$$

Así como existen los conjuntos en la lógica clásica también existen los conjuntos difusos. Si a una proposición P se le asigna un conjunto A; entonces el valor de verdad, denotado por T(p), está dado por:

$$T(p) = \mu_A(x) \text{ donde } 0 \leq \mu_A \leq 1 \quad (2)$$

La anterior ecuación indica que el grado de verdad para la proposición $P: x \in A$ es igual al grado de pertenencia de x en el conjunto difuso A.

El objetivo final de la lógica difusa es brindar bases de conocimiento para el razonamiento de proposiciones imprecisas, por lo cual éste razonamiento es llamado aproximado.

2.1.4. COMPARACIÓN ENTRE LÓGICA CLÁSICA Y LÓGICA DIFUSA

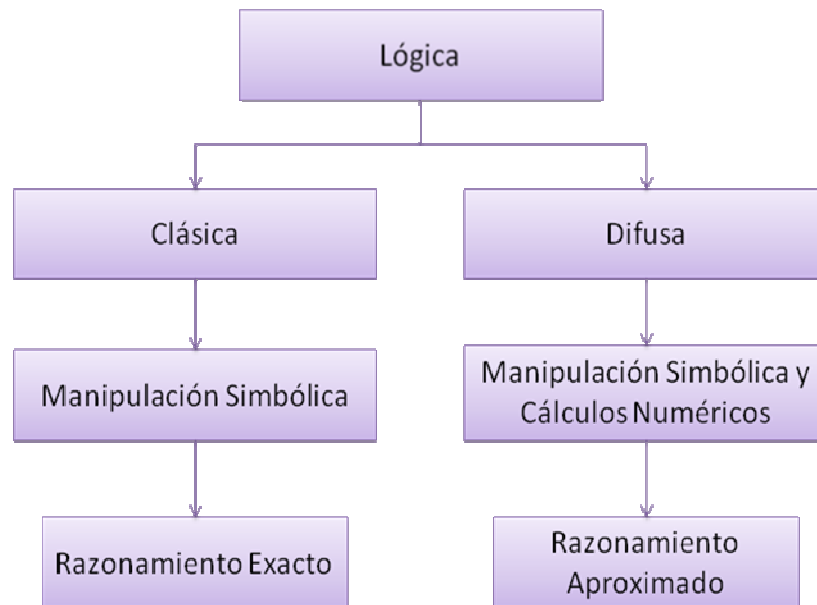


Figura 1. Diagrama conceptual entre la lógica clásica y la lógica difusa.
Tomado de *Descripción General de las Técnicas de Lógica Difusa*⁵.

La diferencia entre estos dos tipos de lógicas radica en que la lógica difusa es bastante flexible en comparación con la clásica pues la primera recurre al uso de los conjuntos difusos, que pueden considerarse generalización de los clásicos; la teoría de conjuntos clásica posee solo dos posibles valores de pertenencia, es decir, pertenece o no pertenece al conjunto, sus elementos solo toman dos valores, 0 ó 1, en cambio, en la teoría de conjuntos difusos existe una pertenencia parcial de un elemento a un conjunto, asignando valores de pertenencia entre 0 y 1. Se concluye que la lógica clásica es precisa, expresa un razonamiento exacto; mientras que la difusa es imprecisa, expresa un razonamiento aproximado.

⁵ http://www.tdr.cesca.es/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf

En un ejemplo que ilustra ésta diferencia; hay un conjunto “*personas viejas*”, entendido desde la lógica clásica, conformado por personas mayores de cierta edad, por ejemplo 60 años, de ésta forma las demás personas con edades inferiores quedan fuera del conjunto, son “*personas jóvenes*”; según lo anterior, una persona de 61 años es vieja y una de 59 años no; sin embargo, no tiene sentido expresar que una persona es vieja y otra no cuando su diferencia de edad son solo 2 años pues no es una discrepancia grande, la persona de 61 años puede ser vieja pero no tan vieja como una de 80 años o al contrario, la persona de 59 años está en el grupo de jóvenes pero es más vieja que alguien de 20 años. Al analizar éste ejemplo desde la lógica difusa, el conjunto “*personas viejas*” no tiene una frontera definida para pertenecer, o no a él, a través de una función de pertenencia se define la transición de joven a viejo, se le asigna a cada valor de edad un grado de pertenencia al conjunto, entre 0 y 1. Entonces, alguien de 59 años pertenece a “*personas viejas*” con un grado de pertenencia 0.7, alguien de 61 con un grado de 0.8, y alguien de 30 años con un grado 0.3. Visto de ésta forma, se concluye que la lógica clásica es un particular de la lógica difusa.

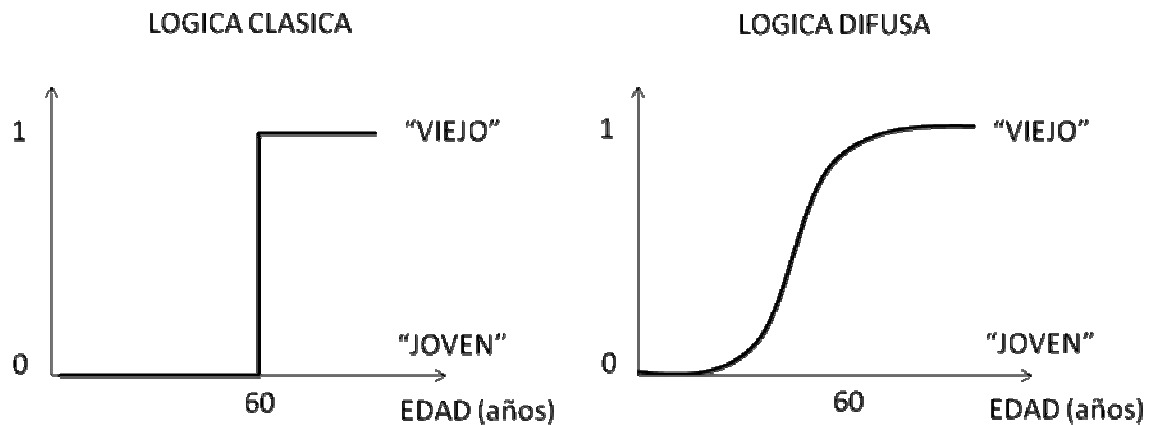


Figura 2. Lógica Clásica Vs. Lógica Difusa.

2.1.5. FUNCIONES DE PERTENENCIA

La forma tradicional de representación de la pertenencia de un elemento a un conjunto, se conoce como función de pertenencia o membresía y se representa así:

$$\mu_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases} \quad (3)$$

Para definir un conjunto difuso se hace uso de una función de pertenencia donde cada valor que toma una variable de entrada, x , la función $\mu_A(x)$ proporciona el grado de pertenencia del valor al conjunto difuso A . La función de pertenencia se define así:

$$\mu_A: X \rightarrow [0,1] \quad (4)$$

En los conjuntos difusos se tiene que X es una agrupación de elementos denotados por x ; el conjunto $A \subseteq X$, se define como un conjunto de pares ordenados tales que:

$$A = \{(x, \mu_A(x)), x \in X\} \quad (5)$$

Donde $\mu_A(x)$ representa la función de pertenencia estableciendo una relación de cada elemento del conjunto A , a un grado de pertenencia entre 0 y 1.

Existen varios tipos de funciones de pertenencia utilizadas dependiendo del criterio a aplicar al problema, la característica que tienen en común es que toman valores de pertenencia entre 0 y 1 con continuidad; se representan de manera gráfica para describir un conjunto difuso y sus parámetros se definen con fórmulas

matemáticas. Las funciones más utilizadas por su facilidad de manejo son Triangular, Trapezoidal, Gaussiana, Campana de Bell y Sigmoidal, que se describen a continuación:

2.1.5.1. TRIANGULAR

Se representa por medio de una forma triangular que se muestra en la figura 3, está compuesta por 3 parámetros a, b, c, para determinar el grado de pertenencia del elemento x.

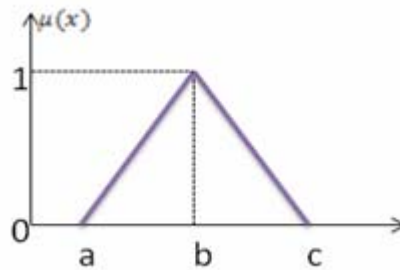


Figura 3. Función de pertenencia Triangular

$$Triangular(x, a, b, c) = \left\{ \begin{array}{ll} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & c \leq x \end{array} \right\} \quad (6)$$

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

2.1.5.2. TRAPEZOIDAL

Se representa por medio de una forma trapezoidal que se muestra en la figura 4, se compone de 4 parámetros a, b, c, d, para determinar el grado de pertenencia del elemento x.

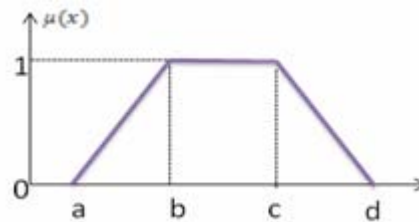


Figura 4. Función de pertenencia Trapezoidal

$$\text{Trapezoidal}(x, a, b, c, d) = \left\{ \begin{array}{ll} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x \end{array} \right\} \quad (7)$$

Los parámetros a, b, c, d, determinan las coordenadas de los 4 ángulos de la función trapezoidal, con $(a, b, c, d) > 0$ y $a < b < c < d$.

2.1.5.3. TIPO CAMPANA DE GAUSS

Se representa por medio de una función gaussiana como se muestra en la figura 5, se compone de 2 parámetros (μ, σ) para determinar el grado de pertenencia del elemento x.

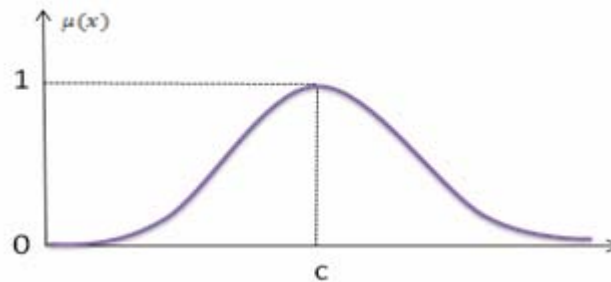


Figura 5. Función de pertenencia Gaussiana

$$\text{Gaussiana}(x, c, \sigma) = e^{-\left[\frac{x-c}{\sigma}\right]^2} \quad (8)$$

El parámetro c representa el centro y σ representa el ancho de la función.

2.1.5.4. TIPO CAMPANA DE BELL

Se representa por medio de la función Campana de Bell como se muestra en la figura 6, se compone de tres parámetros (a , b , c) para determinar el grado de pertenencia del elemento x . Específicamente, se puede ajustar c y a para variar el centro y el ancho de la función y utilizar b para controlar la inclinación de los puntos extremos.

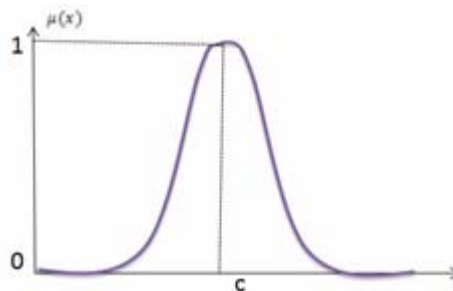


Figura 6. Función de pertenencia Campana de Bell.

$$\text{Bell}(x, a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2+b}} \quad \text{con } b > 0 \quad (9)$$

2.1.5.5. TIPO SIGMOIDAL

Se representa por medio de una figura sigmoïdal como se muestra en la figura 7, se compone de dos parámetros (a, c) para determinar el grado de pertenencia del elemento x.

El parámetro **a** controla la inclinación del punto de transición $x=c$.

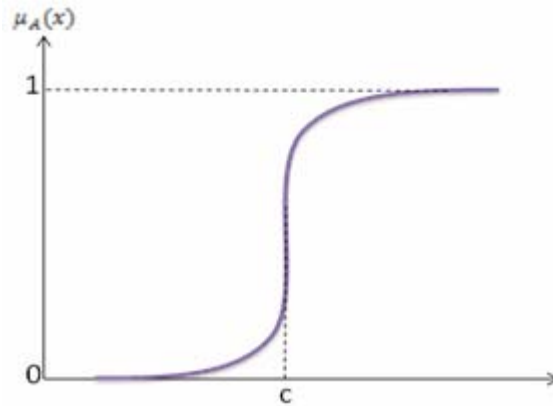


Figura 7. Función de pertenencia Sigmoïdal.

$$\text{Sig}(x, a, c) = \frac{1}{1 + e^{-a(x-c)}} \quad (10)$$

2.1.6. OPERACIONES BÁSICAS CON CONJUNTOS BORROSOS

Las operaciones básicas de los CD⁶ se derivan, de los conjuntos clásicos, existen varias operaciones como son: inclusión, intersección, unión, negación, T-norms, T-Conorms, etc; pero acá solo se expondrán las básicas.

⁶ Conjuntos Difusos

Sea A un conjunto difuso entre 5 y 8, y B un conjunto difuso entorno a 4. Las figuras correspondientes se muestran a continuación:

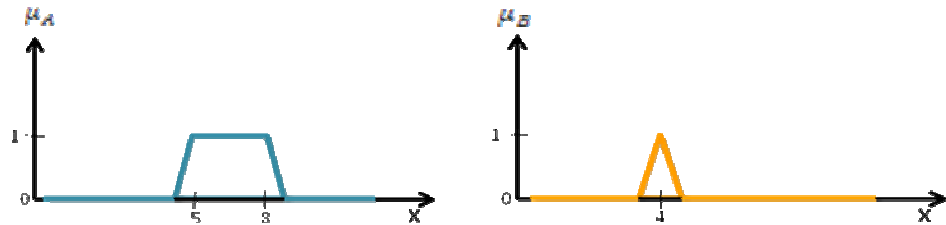


Figura 8. Ejemplos de conjuntos borrosos.

2.1.6.1. INCLUSIÓN O SUBCONJUNTO DIFUSO

Un CD A está contenido en un CD B, si y solo si $\mu_A(x) \leq \mu_B(x)$, para todo $x \in X$,

luego se cumple que:

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x) \quad (11)$$

2.1.6.2. UNIÓN

La unión de los dos CD A y B, es un CD denotado por C, que se describe de la siguiente manera $C = A \cup B$ o $C = A \text{ OR } B$, y donde la función de membresía del

conjunto C; está relacionada por los conjuntos A y B:

$$\mu_c = \text{Max}(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \quad (12)$$

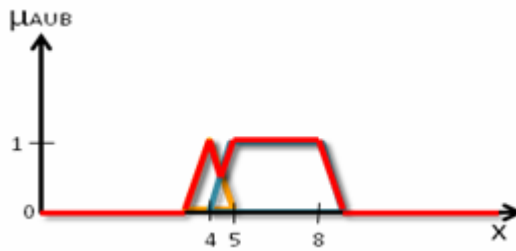


Figura 9. Unión de dos conjuntos borrosos

2.1.6.3. INTERSECCIÓN

La unión de los dos CD A y B, es un CD denotado por C, que se describe de la siguiente manera $C = A \cap B$ o $C = A \text{ AND } B$, y donde la función de membresía del conjunto C; está relacionada por los conjuntos A y B:

$$\mu_C = \text{Min}(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \quad (13)$$

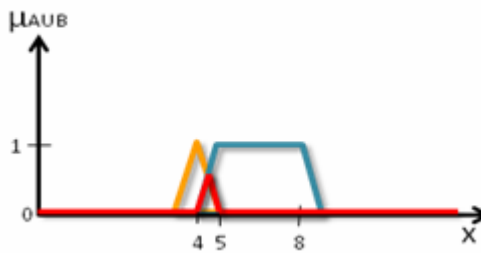


Figura 10. Intersección de dos conjuntos borrosos

2.1.6.4. COMPLEMENTO O NEGACIÓN:

El complemento de un CD A se denomina $\neg A$, NOT A y se define como:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (14)$$

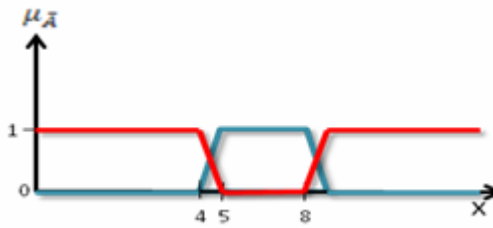


Figura 11. Complemento de un conjunto difuso

2.1.7. VARIABLES LINGÜÍSTICAS

Una variable lingüística se define a través de una “quíntupla” $(x, T(x), X, G, M)$ en la cual:

x es el nombre de la variable lingüística.

T(x) es el conjunto de términos lingüísticos o valores lingüísticos.

X es el conjunto universo.

G es una regla sintáctica por la cual se generan los términos lingüísticos en $T(x)$.

M es una regla semántica la cual asociada con cada término lingüístico A su significado $M(A)$ donde $M(A)$ denota un conjunto difuso en X .

Ejemplo: Sea la variable lingüística “Temperatura”. Podemos definir un conjunto de términos: $T(\text{Temperatura}) = \{\text{Hipotermia, Normal, Fiebre, Fiebre Alta}\}$. Cada término en $T(\text{Temperatura})$ es caracterizado por un conjunto difuso en el universo $[94, 105 \text{ } ^\circ\text{F}]$.

La regla sintáctica se refiere a la forma en que los valores lingüísticos, en el conjunto de términos, son generados. La regla semántica define la función de pertenencia de cada valor lingüístico del conjunto de términos.

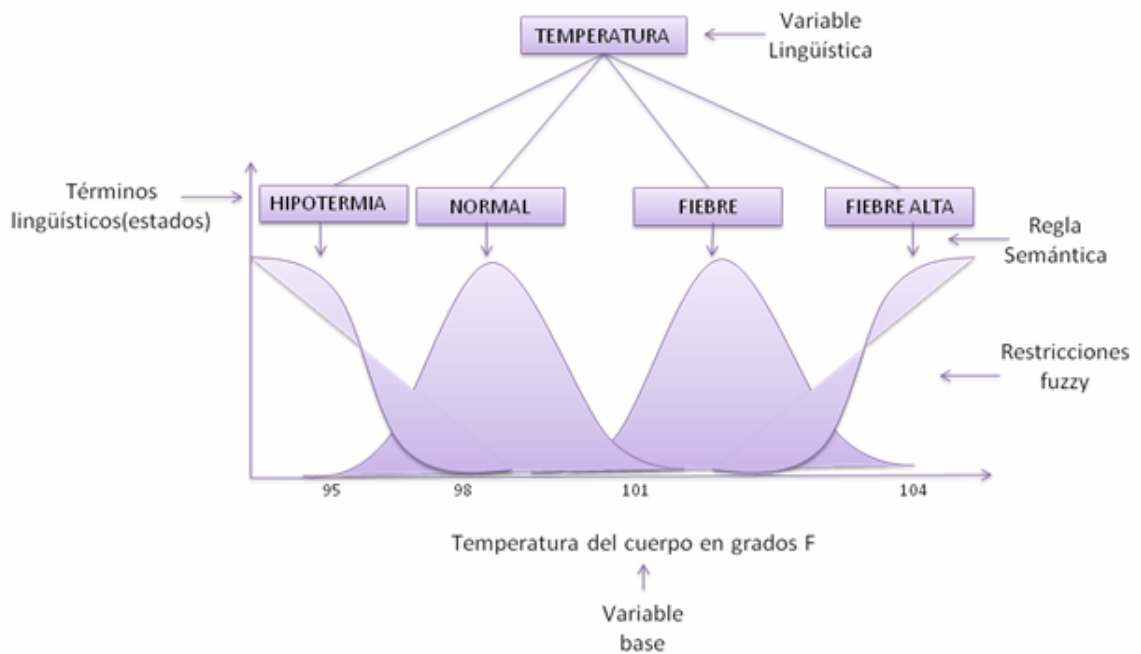


Figura 12. Esquema de los componentes de una variable lingüística.

2.1.8. SISTEMAS BASADOS EN LOGICA DIFUSA

Un sistema basado en lógica difusa está dividido en 3 bloques como se muestra en la figura 13:

Bloque Fusificador: En ésta etapa se le asigna a cada variable de entrada un grado de pertenencia a cada uno de los conjuntos difusos a través de las funciones de pertenencia asociadas a ellos.

Bloque de Inferencia: En este se definen las reglas difusas que componen el sistema o modelo, en este punto es donde los conjuntos difusos de entrada y salida se relacionan a través de ellas; en ésta etapa se obtiene el conjunto difuso de la salida del sistema.

Bloque Defusificador: El objetivo de este punto es que a través de mecanismos matemáticos se encuentre un valor numérico para la variable de salida partiendo del conjunto difuso obtenido en el bloque anterior.

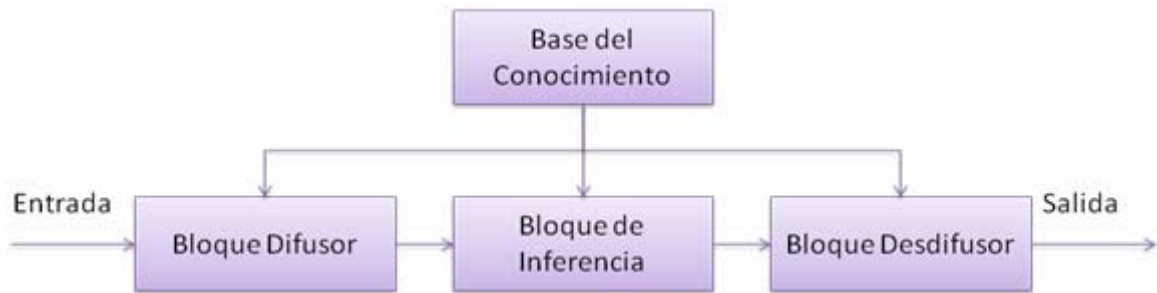


Figura 13. Esquema General de un Sistema Basado en Lógica Difusa.

La clave del buen funcionamiento de un sistema de lógica difusa está en la base del conocimiento, donde se hace uso de las habilidades de un experto en determinada situación para definir de manera apropiada las variables de entrada y salida del sistema así como de las reglas de inferencia, para finalmente obtener los resultados deseados; por esa razón se ve en la figura 13 que la base del conocimiento se relaciona con todos los bloques del sistema.

Existen dos tipos de sistemas de lógica difusa, Mamdani y Sugeno de los que se da una explicación a continuación.

2.1.8.1. SISTEMAS TIPO MAMDANI



Figura 14. Sistema Difuso Mamdani. Proceso general.

En un sistema difuso tipo Mamdani se distinguen las siguientes partes:

2.1.8.1.1. FUSIFICADOR

La entrada de un sistema de lógica difusa tipo Mamdani normalmente es un valor numérico proveniente, por ejemplo, de un sensor; para que este valor pueda ser procesado por el sistema difuso se hace necesario convertirlo a un "lenguaje" que el mecanismo de inferencia pueda procesar, ésta es la función del fusificador, tomar los valores numéricos provenientes del exterior y convertirlos en valores "difusos" para procesarlos con el mecanismo de inferencia. Estos valores difusos son los niveles de pertenencia de los valores de entrada a los diferentes conjuntos difusos en los cuales se ha dividido el universo de discurso de las diferentes variables lingüísticas de entrada al sistema.

2.1.8.1.2. MECANISMO DE INFERENCIA DIFUSA

Teniendo las entradas difusas del fusificador, deben ser procesadas para generar una salida difusa. La tarea del bloque de inferencia es tomar las entradas difusas y en base a las reglas generar una salida difusa.

Las reglas del mecanismo de inferencia difusa son heurísticas y son de la forma “SI (antecedente) ENTONCES (consecuente)”, por ejemplo, SI hace mucho calor ENTONCES disminuir la temperatura.

La inferencia consiste en cuantificar cada premisa a través del operador difuso en el antecedente y/o que corresponden a las operaciones entre conjuntos Min y Max, respectivamente, luego se aplica el consecuente de la regla.

Una regla de la base de conocimiento tiene dos partes, el antecedente y el consecuente como se observa en la figura 15.



Figura 15. Partes de una regla difusa de tipo Mamdani.

En un sistema difuso tipo Mamdani tanto el antecedente como el consecuente de las reglas están dados por expresiones lingüísticas.

2.1.8.1.3. DEFUSIFICADOR

La salida que genera el mecanismo de inferencia es una salida difusa, lo cual significa que no puede ser interpretada por un elemento externo (por ejemplo un controlador) que solo manipule información numérica. Para lograr que la salida del sistema difuso sea numérica, hay que convertirla, éste proceso lo realiza el defusificador.

La salida del mecanismo de inferencia es un conjunto difuso resultante, para generar la salida numérica a partir de estos conjuntos existen varias opciones dos de ellas son: el Centro de Gravedad y los Centros Promediados, se hallan por medio de las Ec. 15 y 16, respectivamente.

$$y = \frac{\sum_i b_i \int \mu(i)}{\sum_i \int \mu(i)} \quad (15) \quad y_h = \frac{\int \mu_{B_m}(S_m) dy}{\int \mu_{B_m}(S_m) dy} \quad (16)$$

2.1.8.2. SISTEMAS TIPO SUGENO

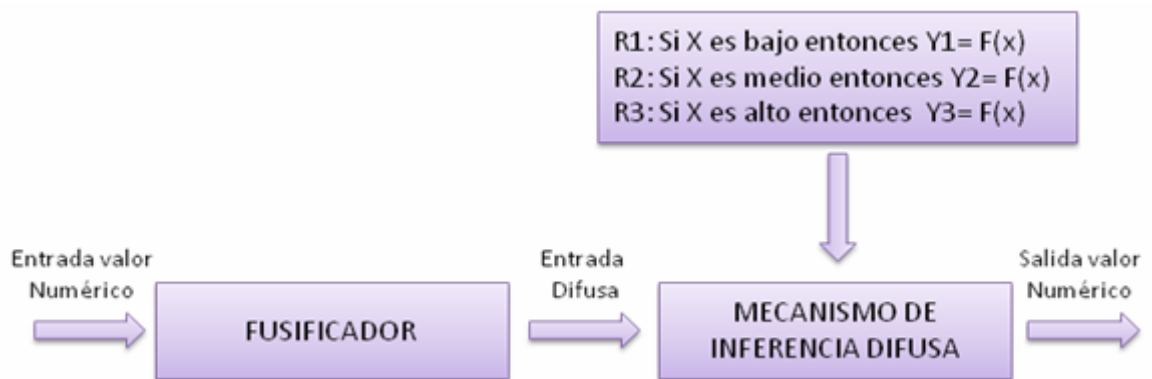


Figura 16. Sistema Difuso Sugeno. Proceso General.

Los sistemas Sugeno están compuestos por dos partes:

2.1.8.2.1. FUSIFICADOR

El fusificador realiza la misma función que en los sistemas mamdani.

2.1.8.2.2. MECANISMO DE INFERENCIA DIFUSA

Su función es tomar las salidas difusas que vienen del fusificador y las convierte en salidas numéricas.

Las reglas de un sistema sugeno son diferentes a las de los sistemas mamdani pues su consecuente ya no es una etiqueta lingüística sino una función de la variable de entrada, de modo que las reglas son de la forma como se muestra en la figura 17.



Figura 17. Partes de una regla difusa de tipo Sugeno.

En los casos en que el sistema Sugeno es de orden cero, la función f de la salida es una constante, en los demás casos por lo general las funciones son lineales.

En los sistemas difusos tipo Sugeno, los valores de salida del mecanismo difuso ya son numéricos por lo que no se necesita una etapa de defusificación.

Para calcular la salida del sistema difuso se halla el promedio ponderado de las salidas numéricas, para un sistema con dos reglas la salida del sistema difuso sería:

$$Y = \frac{W_1 Y_1 + W_2 Y_2}{W_1 + W_2} \quad (17)$$

2.1.9. SISTEMAS DIFUSOS DE DOS CAPAS O NIVELES

El objetivo de los modelos difusos de dos capas es dividir un sistema difuso en dos subsistemas o submodelos para hacerlo sencillo y comprensible, además consume un tiempo de proceso menor que si se hiciera de una sola vez, esto se lleva a cabo tomando las salidas del submodelo 1, llamado capa 1, e introduciéndolas como entradas en la capa o submodelo 2, en ésta capa pueden existir entradas adicionales si la situación así lo requiere.

La razón por la que existen los sistemas difusos de dos capas es que gracias a éstos se pueden procesar muchos más factores que intervienen dentro de un modelo, ya que si todas las variables que afectan un modelo fueran procesadas en una sola capa, esto conllevaría a que se produjeran muchas, complicadas y extensas reglas difusas lo que implicaría un tiempo computacional muy largo, de manera que a través de un sistema difuso de dos niveles o capas se logran procesar modelos mucho más complejos en menor tiempo y de forma más sencilla, por eso la necesidad de un prototipo para crear modelos de éste tipo.

La estructura general de un modelo de dos capas se muestra en la figura 18.

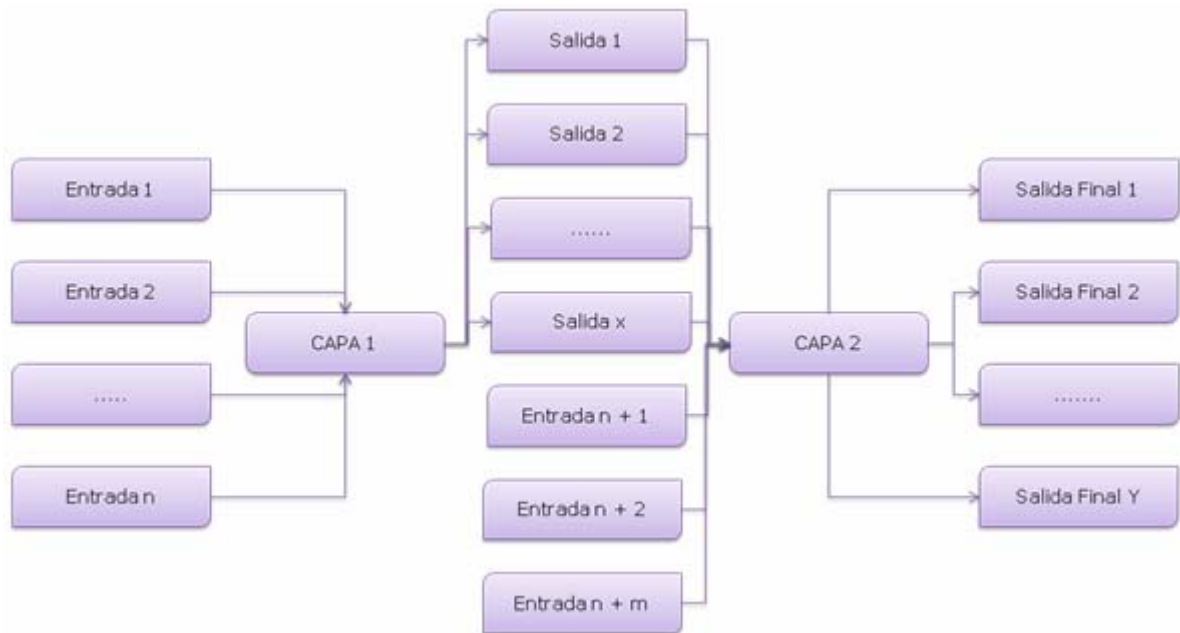


Figura 18. Estructura General de un Modelo Difuso de dos capas.

2.2. APLICACIONES DE LOS SISTEMAS DIFUSOS DE DOS CAPAS

2.2.1. PLANIFICACIÓN DEL DISEÑO DE PLANTAS

2.2.1.1. INTRODUCCIÓN

El objetivo más significativo de cualquier empresa ha sido la máxima utilización de las instalaciones disponibles para alcanzar la meta deseada de productividad y rentabilidad. El diseño de una fábrica se trata de la selección de la mayoría de los ajustes apropiados y efectivos para los departamentos dentro de un plano abierto continuo para permitir una mayor eficiencia de trabajo.

Los requerimientos de espacio para plantas o fábricas y las relaciones laborales dentro de estas son factores importantes que determinan su diseño. Uno de los principales problemas al desarrollar modelos para el diseño de plantas es la

vaguedad natural asociada a los parámetros que los rigen. La tasa del flujo de personal entre los diferentes departamentos, así como otros factores implicados en éste problema deben verse como entradas vagas. A través del presente ejemplo se expone una aplicación de los sistemas fuzzy de dos capas para contribuir a encontrar una solución al diseño de plantas, hallando un índice de cercanía entre facilidades⁷ que da una idea al diseño de que tan cerca se deben ubicar.

2.2.1.2. GENERALIDADES

El objetivo del diseño de una planta es hallar una ordenación de las áreas de trabajo y del equipo, de tal forma que se reduzca el costo de fabricación, se aproveche al máximo el espacio, se facilite la supervisión para los puestos de trabajo que lo requieran, todo esto conlleva a aumentar y mejorar la producción, por ejemplo hablando de una fábrica. La mejor distribución es la que integra a hombres, materiales y maquinaria, así como cualquier otro factor en una gran unidad operativa, es decir que en cierto modo, convierte la planta en una máquina única.

Los factores más importantes para lograr tal fin, en los que se basa éste ejemplo son las variables lingüísticas tenidas en cuenta para la creación del modelo, estas fueron:

Flujo de personal: Una buena distribución prevé los movimientos de los obreros de producción, personal indirecto y supervisores. Los trabajadores deben poder alcanzar las piezas con facilidad, los elementos que bloqueen los movimientos de los hombres o los obliguen a efectuar rodeos, son

⁷ Instalaciones o departamentos de una planta.

obstrucciones que dan como resultado pérdida de tiempo. Los pasillos demasiado estrechos para los obreros de mantenimiento o transportadores de material, conducen a la congestión de los mismos. Un equipo (maquinaria, utillaje) amontonado o junto que no permita un área de acceso para los inspectores y supervisores, da como resultado un control deficiente de las operaciones. Son inconvenientes que deben eliminarse a la hora de diseñar una planta.

Vínculo de supervisión: El diseño de la planta puede influir en la facilidad y calidad de la supervisión. Por ejemplo, una oficina situada en un entresuelo, desde la cual un capataz puede vigilar la planta de trabajo, representa un ahorro de tiempo en cuanto a supervisión. Existen otras soluciones para ubicar los puestos de trabajo de manera que la labor de supervisión sea exitosa; cuando los puestos de trabajo están ordenados en secuencia directa, los supervisores pueden controlar fácilmente el trabajo desarrollado, estos 2 tipos de distribución favorecen el trabajo de supervisión, los demás tipos de distribución pueden conllevar a que ésta función sea difícil.

Vínculo de ambiente: En una planta se debe tener especial cuidado con las condiciones de los materiales que se manejan, así como con la seguridad de los empleados, teniendo en cuenta las condiciones de los materiales y el lugar donde se almacenan y se manipulan, es decir, el ambiente en el que se tratan y la manera como esto afecta a los trabajadores, por tal motivo se debe prever el tipo de materiales que se van a manejar, son fluidos o sólidos, duros o blandos, flexibles o rígidos, etc. Por ejemplo, la pintura que debe usarse a temperatura ambiente, no puede ser almacenada en estantes situados en zonas frías; y en caso de estar guardada en una habitación con calefacción debe mantenerse alejada de los radiadores.

Vínculo de información: Otro de los factores importantes a la hora de realizar el diseño de una planta es éste, se debe planear teniendo presente, cual departamento requiere entrada o salida de información o materiales de otro departamento y dependiendo de la clase de vínculo que exista entre ellos, situarlos más cercanos o más apartados.

El diseño de plantas es un arreglo de actividades, características y espacios, considerando la relación que existe entre ellos. La mayoría de los problemas de diseño de plantas (FLP⁸) son analíticamente formulados acordes a un problema de asignación cuadrático (QAP⁹) con el objetivo de minimizar el costo de transporte o maximizar el índice total de cercanía, un modelo clásico en optimización discreta que trabaja enumerando diferentes configuraciones de diseño hasta que el mejor arreglo se obtiene. Aunque matemáticamente elegante, un QAP es un problema NP-hard¹⁰ que implica ser computacionalmente impráctico para problemas que contengan más de 15 departamentos. Los analistas también se enfocan en resolver éste tipo de problemas a través de métodos heurísticos.

⁸ Facilities Layout Problem. Problema de diseño de plantas.

⁹ Quadratic Assignment Problem. Problema de asignación cuadrático.

¹⁰ NP-hard (nondeterministic polynomial-time hard). Polinomio no determinístico de tiempo duro, muy lento para su procesamiento en una computadora.

2.2.1.3. DESCRIPCIÓN DEL MODELO

La construcción del modelo descrito a continuación se basó en el artículo encontrado en internet “Facilities layout planning based on Fuzzy multiple criteria decision-making methodology”¹¹.

2.2.1.3.1. VARIABLES

El presente ejemplo es la presentación de una solución al problema del diseño de plantas con base en un sistema de inferencia difuso de dos capas (FIS) arrojando un índice de cercanía que facilita el proceso de diseño; las variables de entrada de la capa 1 que se asumen para el presente problema, debido a que son las más utilizadas por los expertos, son: nivel de flujo de personal (PF), vínculo de supervisión (VS), condición de trabajo o vínculo de ambiente (VA) y vínculo de información (VI).

Además, se incluyen los pesos de cada uno de éstos parámetros debido a que de ésta forma se da un peso o importancia cada variable en la salida del modelo, estos pesos son calculados por los expertos en el tema. Los pesos introducidos en el modelo son: Peso del flujo de personal (PFP), Peso del vínculo de supervisión (PVS), Peso del vínculo ambiental (PVA) y el Peso del vínculo de información (PVI).

Los valores asociados con las variables de entrada son:

Flujo de personal: Muy Bajo, Bajo, Medio, Alto, Muy Alto.

¹¹ Planeación del diseño de plantas basado en la metodología de toma de decisiones de multicriterios difusos.

Vínculo de supervisión: Insignificante, Considerable, Moderado, Esencial, Muy esencial.

Vínculo ambiental: Muy Seguro, Seguro, Inseguro, Riesgoso, Muy Riesgoso.

Vínculo de información: Muy Débil, Débil, Medio, Fuerte, Muy Fuerte.

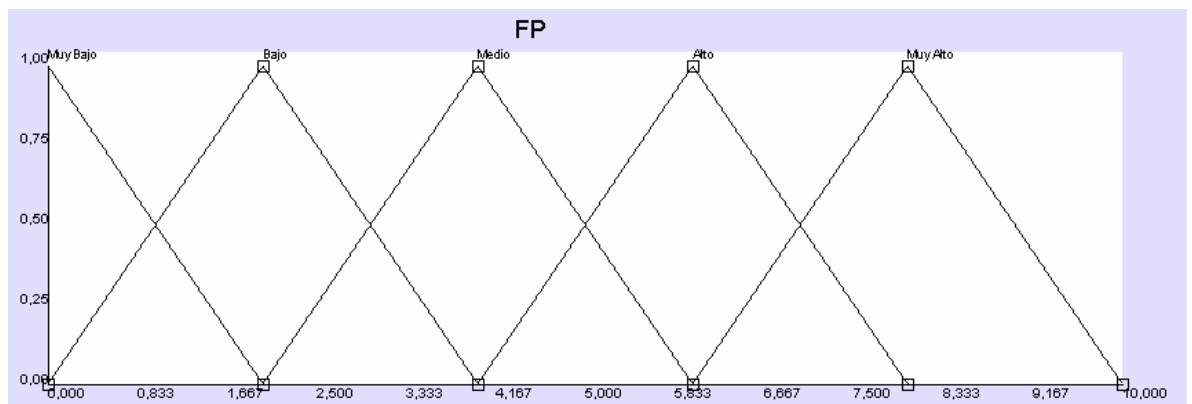
Peso del Flujo de Personal: Bajo, Medio, Alto.

Peso del Vínculo de Supervisión: Bajo, Medio, Alto.

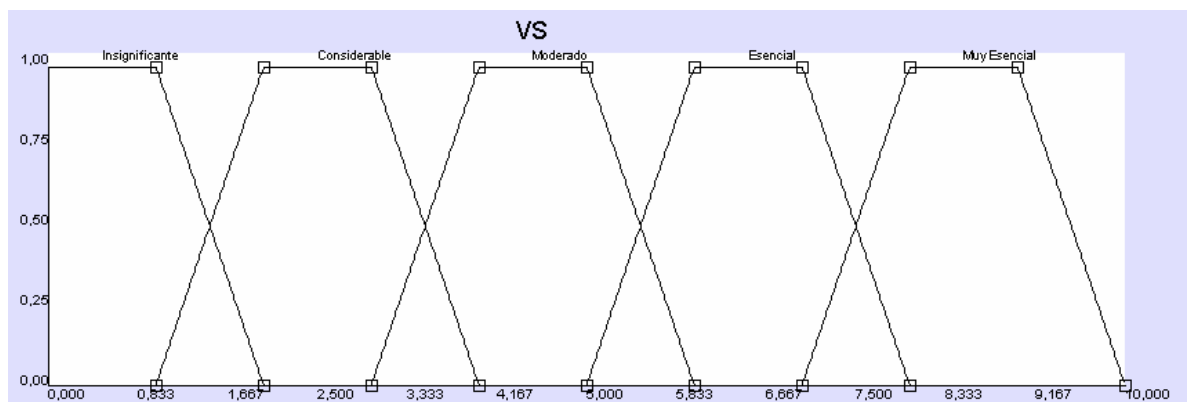
Peso del Vínculo Ambiental: Bajo, Medio, Alto.

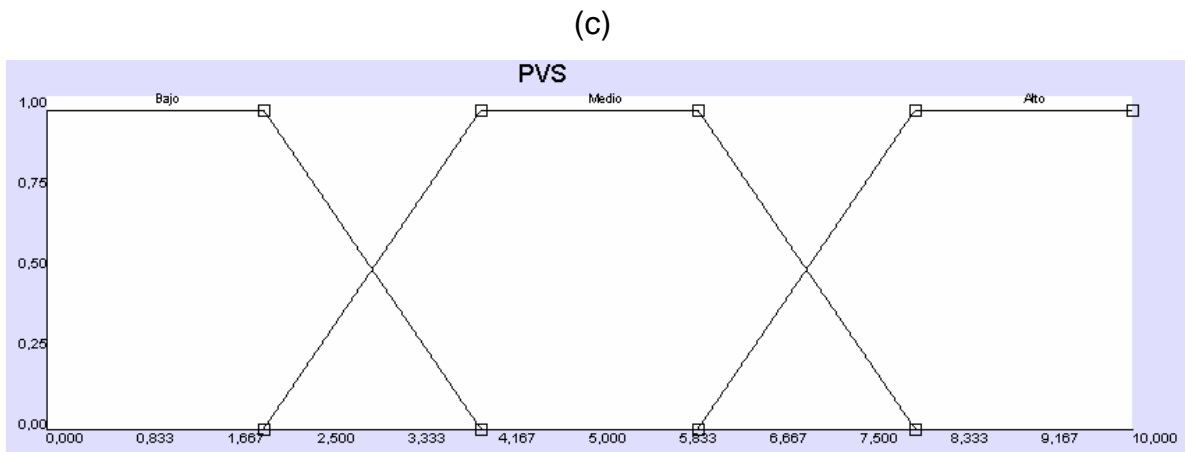
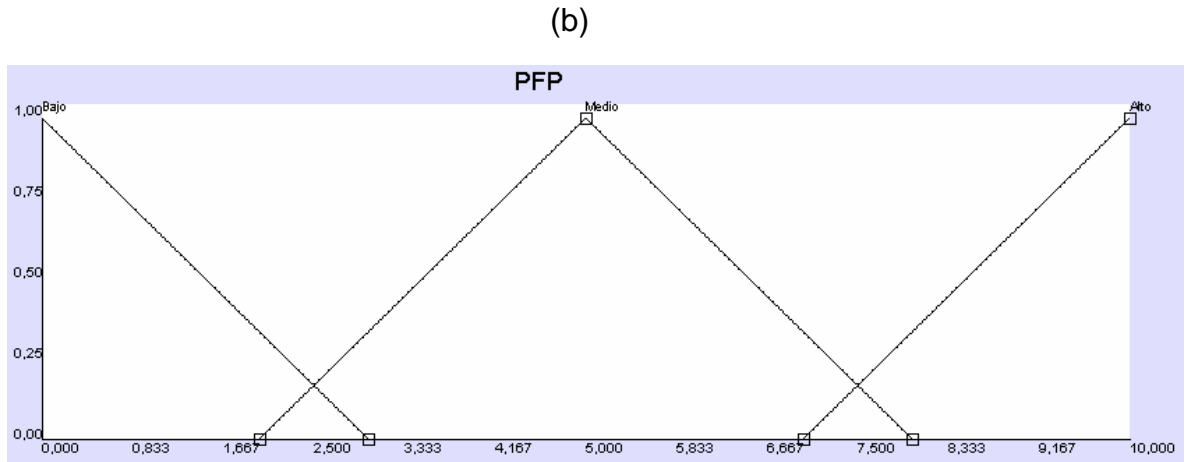
Peso del Vínculo de Información: Bajo, Medio, Alto.

Los valores de las variables lingüísticas de entrada son considerados dentro de una escala designada de [0, 10]. Los conjuntos de las variables de entrada se muestran en la figura 19.



(a)





(d)

Figura 19. Variables de Entrada Modelo Diseño de Plantas (Capa 1).
 (a) Flujo de Personal (b) Vínculo de Supervisión, la estructura de las variables Vínculo Ambiental y Vínculo de Información es la misma. (c) Peso del Flujo de Personal. (d) Peso del Vínculo de Supervisión, la estructura de los pesos de Vínculo Ambiental y Vínculo de Información es la misma.

La función de la capa 1 es determinar la resultante de la relación entre un parámetro y su peso, lo que indica la importancia de éste en la salida del modelo.

Las salidas de la capa 1 se toman como entradas en la capa 2, éstas variables de salida están dentro de una escala [0, 10] y los valores asociados a ellas son:

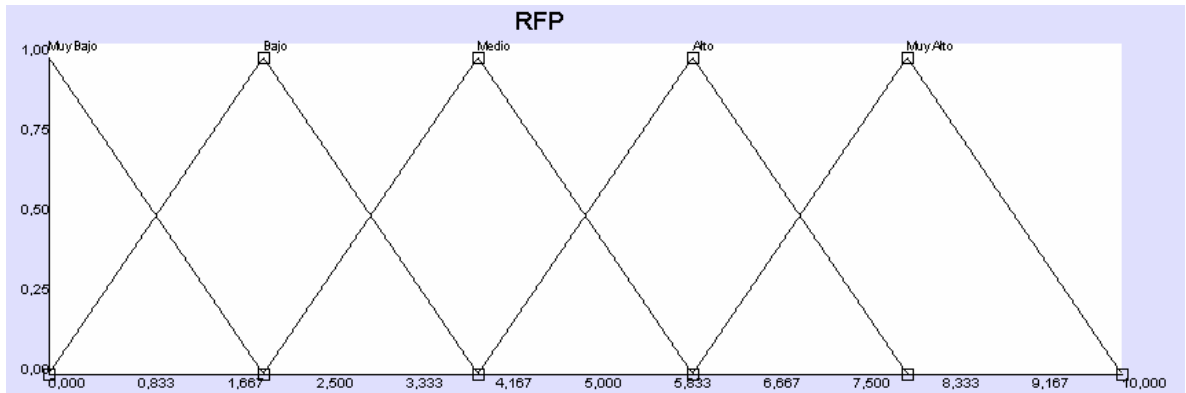
Resultante del Flujo de Personal: Muy Bajo, Bajo, Medio, Alto, Muy Alto.

Resultante del Vínculo de Supervisión: Muy Bajo, Bajo, Medio, Alto, Muy Alto.

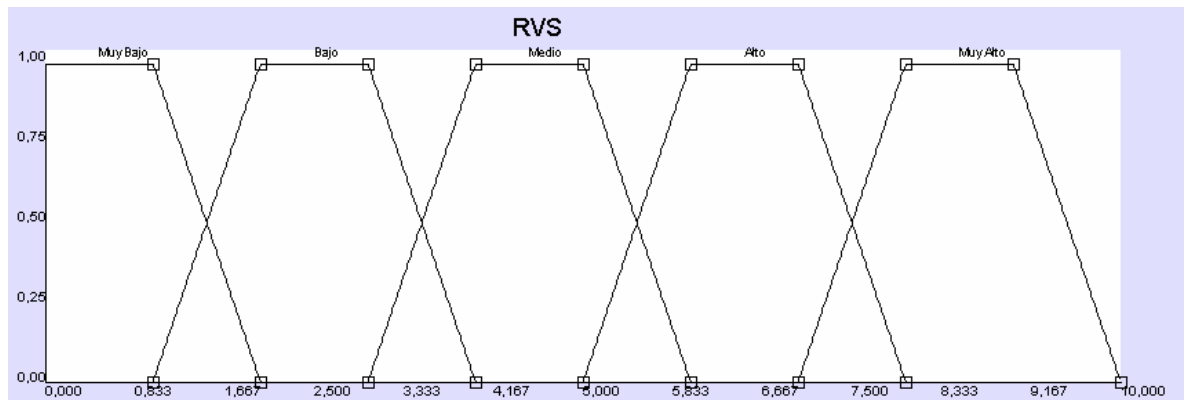
Resultante del Vínculo Ambiental: Muy Bajo, Bajo, Medio, Alto, Muy Alto.

Resultante del Vínculo de Información: Muy Bajo, Bajo, Medio, Alto, Muy Alto.

Los conjuntos de las variables de salida se muestran en la figura 20.



(a)



(b)

Figura 20. Variables de Salida Capa 1 y Entrada Capa 2 del Modelo Diseño de Plantas.

(a) Resultante del Flujo de Personal. (b) Resultante del Vínculo de Supervisión, las resultantes de las variables Vínculo ambiental y Vínculo de Información se representan de la misma manera.

La salida de la capa 2 mide la clasificación del índice de cercanía¹² entre dos facilidades, con valores estándar asociados Innecesario, Insignificante, Ordinario, Importante, Esencial y Muy Esencial, dentro del universo de discurso [0,6] como se muestra en la figura 21.

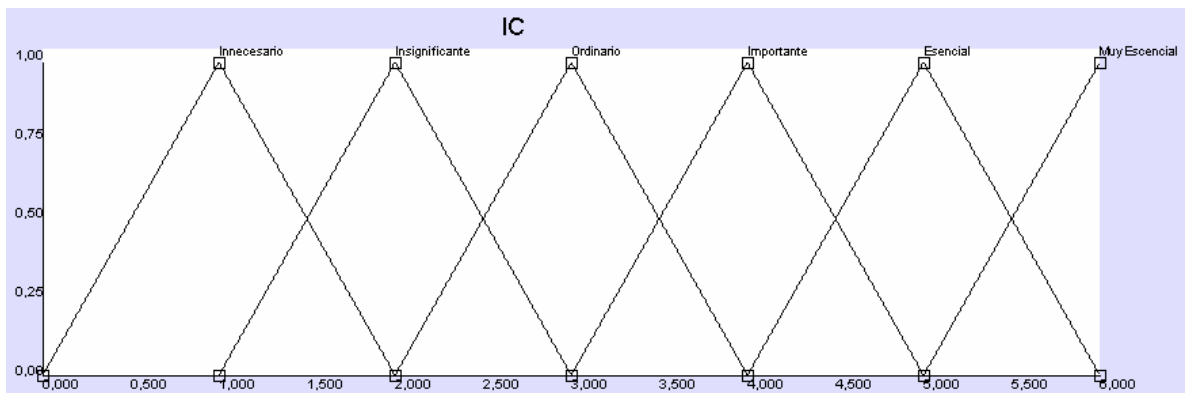


Figura 21. Índice de Cercanía.

Las variables asociadas al modelo, sus universos de discurso, sus conjuntos difusos y sus funciones de pertenencia fueron desarrollados según el conocimiento existente y la lectura del artículo “*Facilities layout planning based on Fuzzy multiple criteria decision-making methodology*”¹³.

¹² Grado de importancia de la cercanía que debe existir entre dos departamentos. Cuanto mayor es el índice mayor debe ser la cercanía, por lo tanto la distancia entre los departamentos debe ser menor.

¹³ S. K. Deby and B. Bhattacharyya. Production Engineering Department, Jadavpur University, Kolkata, India. Vol.41, no.18, 4487–4504. 2003.

El esquema general del modelo difuso de dos capas se muestra en la figura 22.

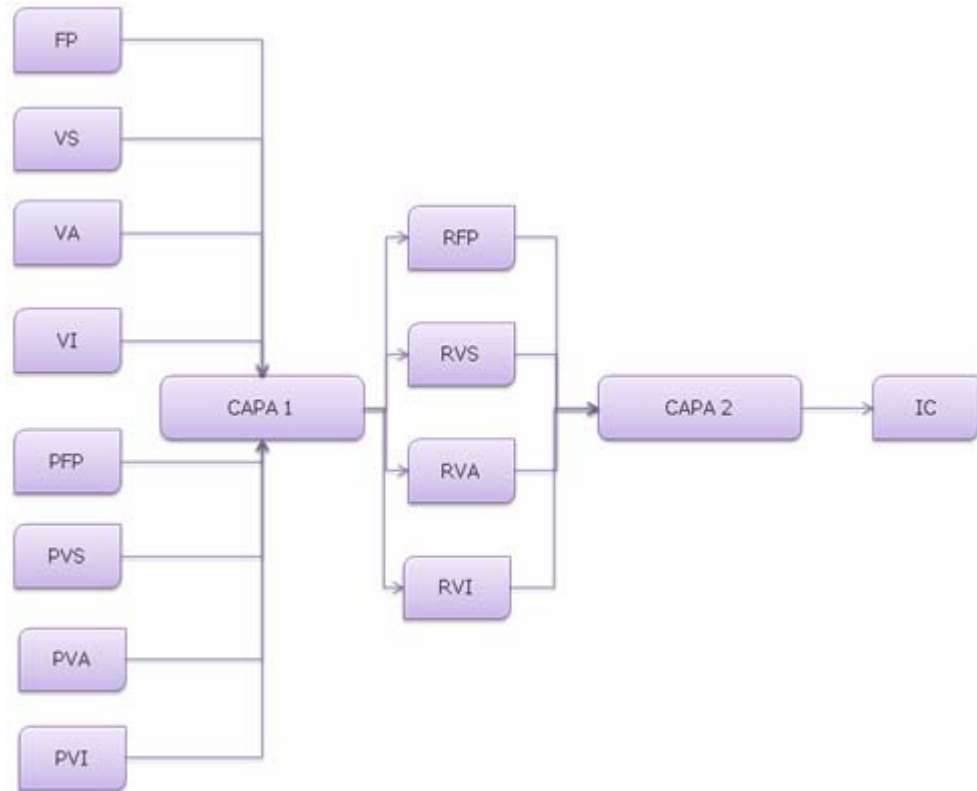


Figura 22. Modelo Difuso de dos capas para la planificación de plantas.

2.2.1.3.2. REGLAS DE DECISIÓN DIFUSAS

Las reglas del modelo fueron creadas a partir del artículo mencionado anteriormente y según algunas lecturas¹⁴ hechas por las autoras. Aquí se presentan algunas:

¹⁴ GRANT IRESON, William. Planeamiento de fábricas. Barcelona, España. EDITORIAL HISPANO EUROPEA. 1966.; MUTHER, Richard. Distribución en planta. Barcelona, España. EDITORIAL HISPANO EUROPEA. 1965.

Si el flujo de personal relacionado entre dos facilidades es muy alto, entonces ellas deben ser localizadas cerca la una de la otra, entonces el índice de cercanía debe ser Muy Esencial.

Si la relación de supervisión entre dos facilidades es muy alta, entonces deben ser localizadas cerca la una de la otra, entonces el índice de cercanía debe ser Muy Esencial.

Si la relación ambiental entre dos departamentos es muy arriesgada entonces ellos deben ser localizados lejos el uno del otro, entonces el índice de cercanía es Insignificante.

Si la relación de información entre dos facilidades es muy fuerte, entonces deben ser localizadas cerca la una de la otra. entonces, el índice de cercanía es Muy esencial.

2.2.2. DETERMINACIÓN DE LA PROBABILIDAD DE PICOS TRANSVALVULARES EN ECOCARDIOGRAMAS

2.2.2.1. INTRODUCCIÓN

A menudo cuando los médicos especialistas en cardiología necesitan realizar diagnósticos acerca del funcionamiento del corazón y del sistema circulatorio se hace necesaria la práctica de algunos exámenes que muestren el estado de los mismos.

Existe un estudio que explora la hemodinámica¹⁵ de la sangre, espectrograma doppler¹⁶ y otro que registra gráficamente la actividad eléctrica del corazón, electrocardiograma (ECG¹⁷), éstos fueron empleados por unos investigadores de la universidad La Trobe, en Australia, para seguir el comportamiento del flujo sanguíneo a través de las válvulas cardíacas mitral y sigmoidea aortica, flujo interno diastólico trans-mitral (MDI) y flujo externo sistólico trans-aórtico (ASO). Su objetivo era realizar un análisis hemodinámico de éstas válvulas para obtener de forma automática los puntos de inicio y fin de una onda de los flujos trans-valvulares, en base a un sistema, de manera que no se requiriera la ayuda manual de un experto en lectura de espectrogramas y electrocardiogramas para llevar a cabo dicha labor. El sistema se fundamenta en un modelo difuso de dos capas que según valores extraídos de un ECG y de las curvas de potencia de retrodispersión (bpc¹⁸) doppler del espectrograma, se especifica cuáles de los picos en esas gráficas representan abertura o cierre de las válvulas para luego establecer unos patrones del flujo sanguíneo y por último determinar los tiempos de inicio y fin de una onda.

2.2.2.2. GENERALIDADES

El objetivo del sistema es determinar los puntos de inicio y fin de una onda de los flujos trans-valvulares, tarea que resulta un poco más sencilla para un experto en ultrasonografías, ya que él por inspección visual o cualitativa puede estimar adecuadamente esos puntos, pero los autores del artículo "*Investigation of a*

¹⁵Estudio de la dinámica de la sangre y flujo sanguíneo dentro del corazón y el sistema vascular.

¹⁶ Examen que utiliza ultrasonido para examinar el flujo sanguíneo.

¹⁷ El electrocardiograma (ECG/ECG, del alemán *Elektrokardiogramm*) es el gráfico que se obtiene con el electrocardiógrafo para medir la actividad eléctrica del corazón en forma de cinta gráfica continua.

¹⁸ Son las curvas según las cuales se analiza el espectrograma doppler.

*Hierarchical Fuzzy Model for Echocardiographic Assessment of Trans-valvular Hemodynamic Indexes*¹⁹ quisieron diseñar un sistema que desempeñara esa función automáticamente.

El sistema general se divide en cuatro etapas de análisis como se muestra en la figura 23. Primero, la extracción de parámetros, se forman vectores de parámetros de las características del ECG y del espectrograma que influyen en el resultado esperado, la amplitud, la pendiente y la duración de un punto. Segundo, un sistema fuzzy de dos capas se usa para analizar los puntos de muestreo, de las curvas de entrada, candidatos para ser picos que representen abertura o cierre de las válvulas, verificando que tengan la forma, fuerza y los tiempos de duración adecuados. En esta fase, el uso considerable de información del contexto de tiempo presente en el ECG ayuda en la detección de puntas de abertura y cierre valvular. En la tercera etapa, un patrón de clasificación es aplicado a las salidas obtenidas de la segunda etapa. El sistema clasifica los flujos dentro de diferentes patrones de acuerdo a la presencia de puntas de apertura o cierre de las válvulas. En la última fase, los tiempos de inicio y fin de los flujos MDI y ASO son finalmente determinados usando reglas particulares para cada patrón.

¹⁹ Jie Gong, Richard Kirsner, Andrew Maclsaac, Carol Drossos and James Cameron. Department of Electronic Engineering, La Trobe University, Bundoora, Australia. Cardiac Investigation Unit, St Vincent's Hospital, Melbourne, Australia. 2002.

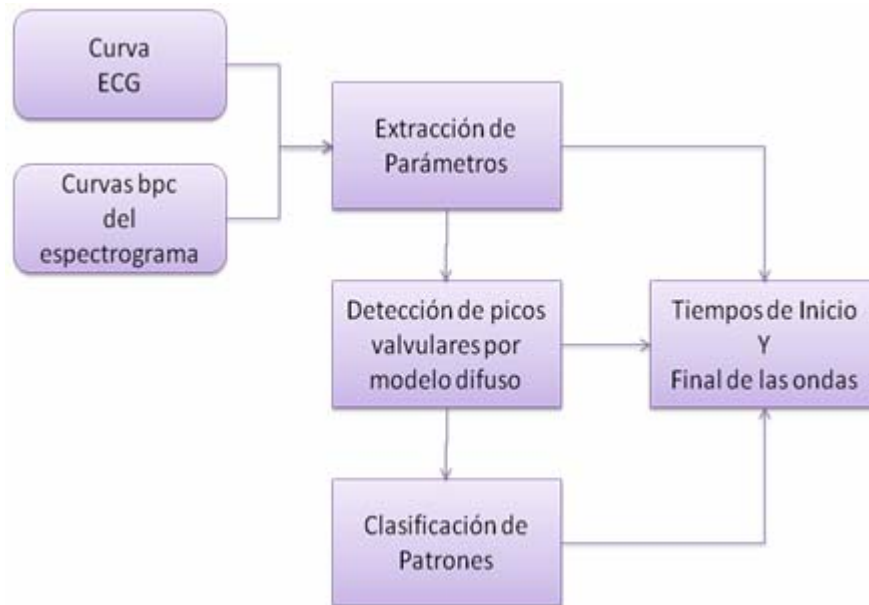


Figura 23. Esquema General del Sistema de Detección de Tiempos de Inicio y Fin de Ondas de Flujo Trans-valvular.

A continuación se describirá solamente la segunda etapa del sistema puesto que es en la que interviene el modelo difuso.

2.2.2.3. DESCRIPCIÓN DEL MODELO DIFUSO DE DOS CAPAS

2.2.2.3.1. VARIABLES

El modelo que se describe a continuación se basa en algunas características extraídas del artículo mencionado en el numeral anterior, pero otros de sus rasgos fueron definidos por las autoras del presente documento, según lecturas realizadas acerca de la actividad eléctrica del corazón y el comportamiento de los flujos sanguíneos dentro del mismo.

Existen dos modelos, uno para el flujo trans-mitral y el otro para el flujo trans-aortico, la estructura de los dos es la misma, la única variante entre ellos es el planteamiento de sus reglas, algunas de ellas se mostrarán más adelante. Para la primera capa se cuenta con cuatro variables de entrada y dos de salida, las variables de entrada son: La amplitud de un pico en el ECG (amplitud_E), la pendiente de un pico en el ECG (pendiente_E), la amplitud de un pico en el espectrograma (amplitud_bpc), la pendiente de un pico en el espectrograma (pendiente_bpc), los conjuntos difusos de estas variables poseen funciones de pertenencia de tipo sigmoïdal, como se muestra en la figura 24.

Los valores asociados a las variables lingüísticas de entrada son como sigue:

Amplitud_E: Negativa Alta, Negativa Baja, Positiva Baja, Positiva Alta.

Pendiente_E: Negativa Alta, Negativa Baja, Positiva Baja, Positiva Alta.

Amplitud_bpc: Negativa Alta, Negativa Baja, Positiva Baja, Positiva Alta.

Pendiente_bpc: Negativa Alta, Negativa Baja, Positiva Baja, Positiva Alta.

Las variables de salida corresponden a la probabilidad de que alguno de esos puntos de entrada represente un pico candidato a ser pico de abertura o cierre, probabilidad de pico en el ECG (P_PicoE), probabilidad de pico en el espectrograma (P_PicoBpc), los conjuntos difusos de estas variables poseen funciones de pertenencia de tipo campana de bell, como se muestra en la figura 25 (a).

Los valores asociados a las variables lingüísticas de salida de la capa 1 son como sigue:

P_PicoE: Muy Alta, Muy Baja.

P_PicoBpc: Muy Alta, Muy Baja.

La función de la capa 1 es determinar dos probabilidades, que un pico sea candidato a representar un pico de abertura o cierre en el ECG, o que un pico sea candidato a representar un pico de abertura o cierre en el espectrograma. Para las reglas de la primera capa solo se relacionaron amplitud_E y pendiente_E para hallar las probabilidades de picos en el ECG y para hallar las probabilidades de picos en el espectrograma se relacionaron solamente amplitud_bpc y pendiente_bpc.

Como ya se sabe, las variables de salida de la capa 1 son las variables de entrada de la capa 2, por lo tanto, para la capa 2 las variables de entrada son P_PicoE y P_PicoBpc, además de éstas dos existe otra llamada Duración, que representa el tiempo que dura cierto pico, es una variable importante a la hora de establecer si el pico encontrado representa la abertura o el cierre de la válvula. Los conjuntos de la variable duración se muestran en la figura 25 (b).

Los valores asociados a la variable lingüística Duración son: Baja, Media y Alta. Las variables de salida de la capa 2 son: PP_aberturaE y PP_cierreE, probabilidad que un pico represente la abertura o el cierre de la válvula en el ECG, respectivamente, PP_aberturaBpc y PP_cierreBpc, probabilidad que un pico represente la abertura o el cierre de la válvula en el espectrograma,

respectivamente, los conjuntos difusos de éstas variables poseen funciones de pertenencia campana bell como se muestra en la figura 26.

Los valores asociados a las variables lingüísticas de salida de la capa 2 son como sigue:

PP_aberturaE: Baja, Media, Alta.

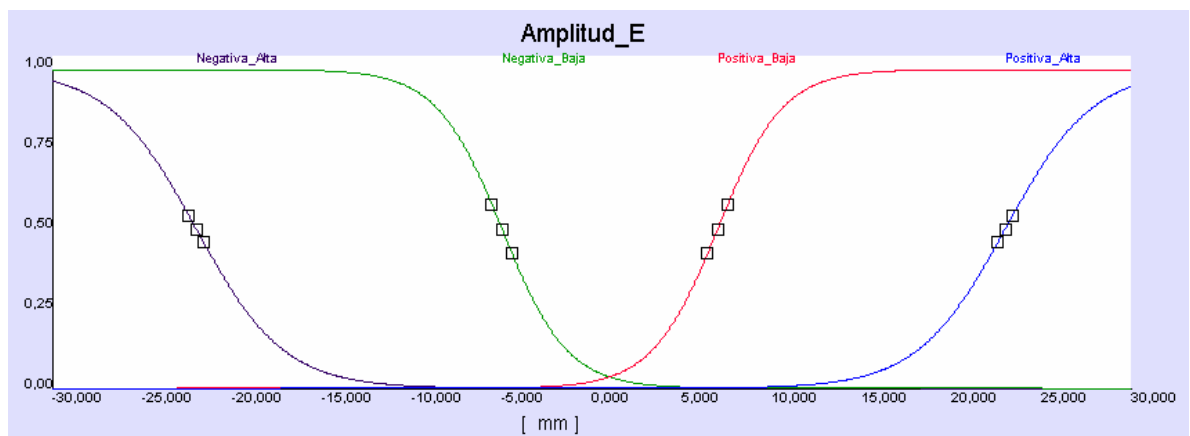
PP_cierreE: Baja, Media, Alta.

PP_aberturaBpc: Baja, Media, Alta.

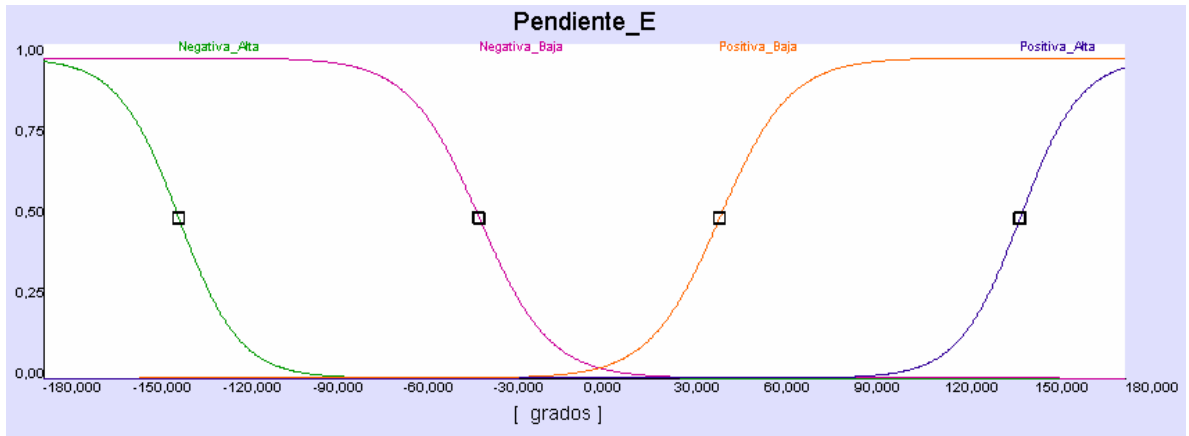
PP_cierreBpc: Baja, Media, Alta.

La función de la capa 2 es entonces definir qué tan probable es que un pico sea de abertura o cierre en la válvula ya sea en el ECG o en el espectrograma.

El esquema general del modelo se muestra en la figura 27.



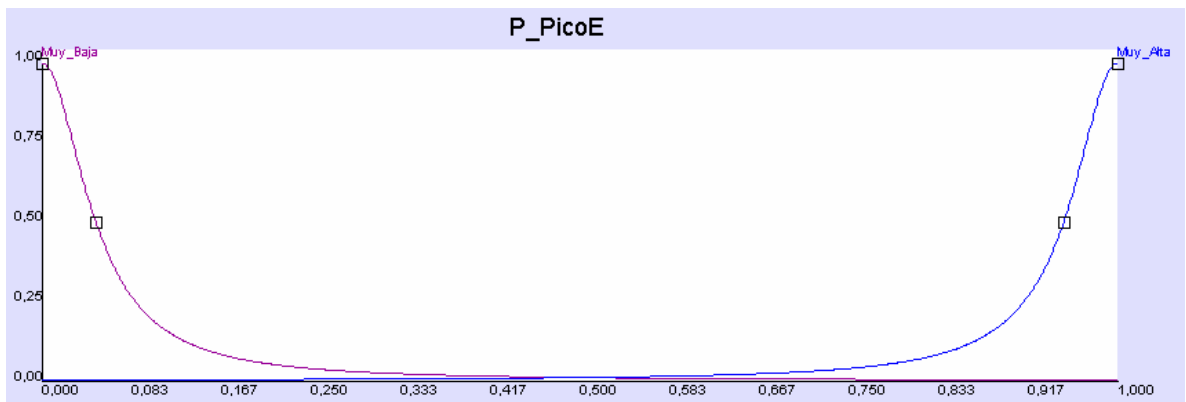
(a)



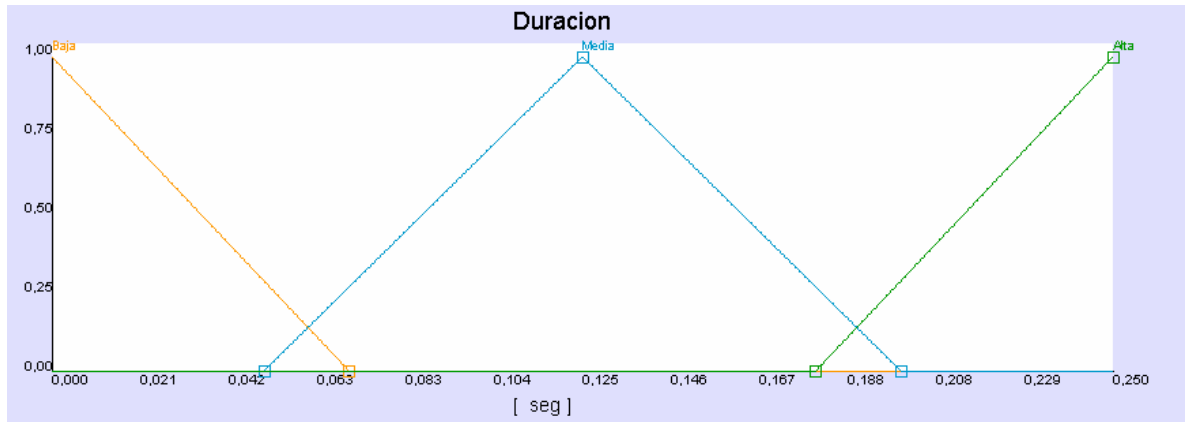
(b)

Figura 24. Variables de Entrada Modelo Probabilidad de Picos en Flujo Trans-Mitral y Trans-Aortico.

(a) Variables amplitud_E y amplitud_bpc, las dos tienen la misma estructura. (b) Variables pendiente_E y pendiente_bpc, las dos tienen la misma estructura.



(a)



(b)

Figura 25. Variables de la Capa 2 Modelo Probabilidad de Picos en Flujo Trans-Mitral y Trans-Aortico.

(a) Variables de salida capa 1 y entrada capa 2, P_PicoE y P_PicoBpc, las dos tienen la misma estructura. (b) Variable de entrada de la capa 2, Duración.

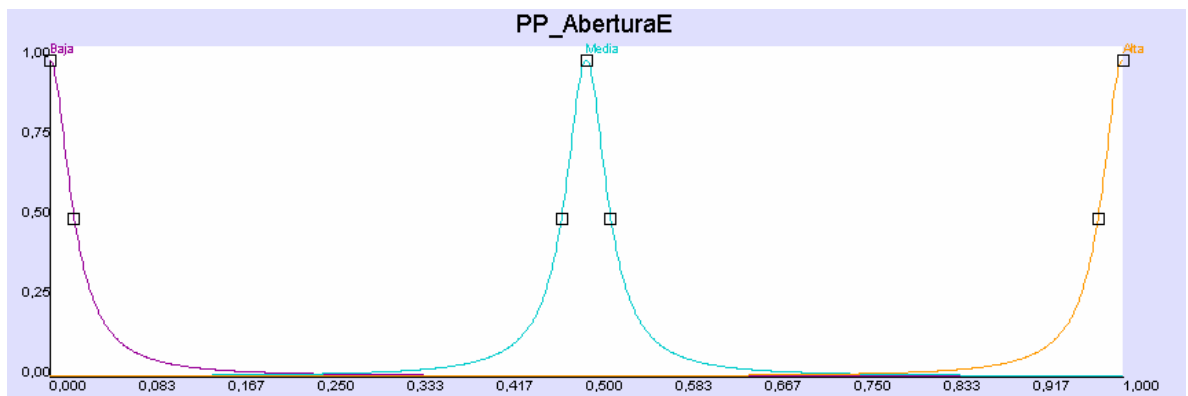


Figura 26. Variables de Salida de la capa 2, PP_AberturaE, PP_CierreE, PP_AberturaBpc, PP_CierreBpc.

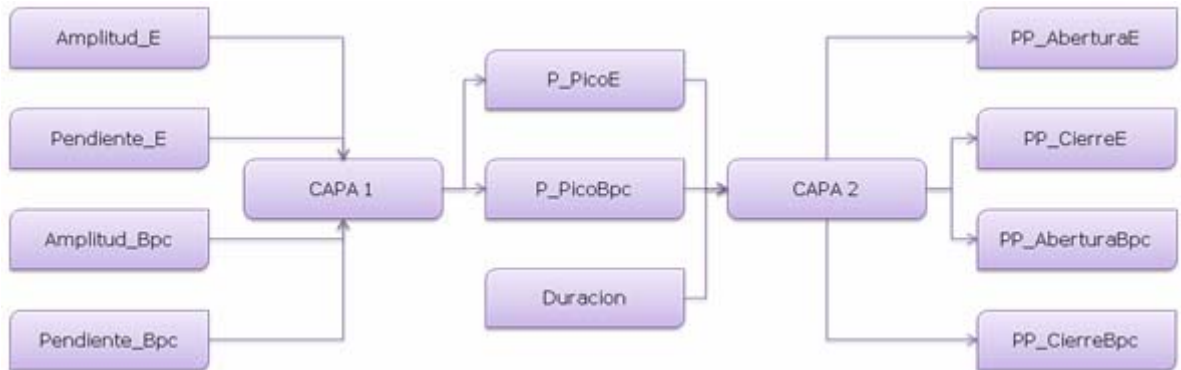


Figura 27. Esquema General Modelo Difuso.

2.2.2.3.2. REGLAS

Las reglas fueron una de las características que las autoras del presente documento desarrollaron en base a artículos encontrados en internet acerca de la actividad eléctrica del corazón y el comportamiento de los flujos sanguíneos dentro del mismo. La probabilidad que un pico represente la abertura o cierre de alguna de las válvulas mitral o aortica, depende esencialmente de 3 factores, amplitud, pendiente o inclinación y duración del pico, a continuación se presentan algunas de las reglas para el modelo del flujo trans-mitral y otras del modelo del flujo trans-aortico:

MODELO FLUJO TRANS-MITRAL

- Cuando la amplitud de un pico es positiva baja y su pendiente es positiva baja existe una probabilidad muy alta que ese punto pico sea uno que represente el cierre de la válvula mitral, pero lo que define si lo es o no, en la segunda capa, es el tiempo de duración del mismo, si esa duración es media entonces la probabilidad que ese pico sea del cierre de la válvula es bastante alta.

- Por el contrario, si la amplitud de un pico es positiva alta y su pendiente es negativa alta existe una probabilidad muy baja que ese punto pico sea uno que represente el cierre de la válvula mitral, por lo tanto, esos picos que presentan probabilidades bajas no se tienen en cuenta en la capa 2, pues los resultados de la capa 1 demuestran que su probabilidad de representar un pico de abertura o cierre es demasiado baja.

MODELO FLUJO TRANS-AÓRTICO

- Cuando la amplitud de un pico es positiva alta y su pendiente es negativa baja existe una probabilidad muy alta que ese punto pico sea uno que represente la abertura de la válvula aortica, pero lo que define si lo es o no, en la segunda capa, es el tiempo de duración del mismo, si esa duración es media entonces la probabilidad que ese pico sea del cierre de la válvula es bastante alta.
- Por el contrario, si la amplitud de un pico es negativa alta y su pendiente es positiva alta existe una probabilidad muy baja que ese punto pico sea uno que represente la abertura de la válvula mitral, por lo tanto, esos picos que presentan probabilidades bajas no se tienen en cuenta en la capa 2, pues los resultados de la capa 1 demuestran que su probabilidad de representar un pico de abertura o cierre es demasiado baja.

2.3. METODOLOGÍA DE DESARROLLO APLICADA

Para el desarrollo de este proyecto es necesario establecer una metodología que permita alcanzar los objetivos planteados y los resultados contenidos en el

cronograma, de igual forma, transformar los requisitos de un usuario en un sistema o herramienta conformada por una serie de interfaces bien definidas que permitan la interacción de las dos partes involucradas.

Actualmente existen diversos tipos de metodologías enfocadas en el desarrollo de proyectos software como los Modelos Secuenciales: desarrollo rápido de aplicaciones DRA, cascada pura, cascada con modificaciones; y Modelos Evolutivos: prototipado simple, prototipado evolutivo, modelo en espiral y proceso unificado, siendo este ultimo el escogido para llevar a cabo el proyecto.

2.3.1. PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

El enfoque metodológico del proceso unificado de Rational (RUP) es un proceso de Ingeniería del Software orientado a objetos que define un enfoque para el desarrollo de software con el objetivo de asegurar la calidad dentro de una serie de recursos de plazo y presupuesto.

La metodología RUP pretende hacer énfasis en los siguientes aspectos:

Es un proceso configurable, por lo tanto se puede ajustar a las características específicas de un proyecto en cuanto a tamaño y complejidad.

Los modelos creados en las distintas actividades utilizan de forma general la notación UML.

Se basa en las siguientes practicas:

- **Control de Cambios de Software:** proporciona guías para controlar y seguir los cambios que se realicen durante el desarrollo.
- **Verificación de la Calidad:** se debe revisar la calidad del sistema con respecto a los requisitos de funcionalidad, fiabilidad y rendimiento.
- **Gestión de Requisitos:** define un proceso de gestión de requisitos con el cual se pueda capturar, organizar, documentar y realizar el seguimiento de los requisitos del sistema y permite su trazabilidad con otros modelos generados durante el desarrollo.
- **Desarrollo Iterativo:** cada iteración representa un periodo de tiempo en el que se realizan diferentes actividades de desarrollo con el objeto de disponer al final de la misma versión del sistema.

La metodología mencionada se llevara a cabo en función de dos dimensiones:

2.3.2. DIMENSIÓN TEMPORAL Y DIMENSIÓN ESTÁTICA

2.3.2.1. DIMENSIÓN TEMPORAL

Representa el aspecto dinámico del sistema, el ciclo de vida se divide en períodos de tiempo en los que se trabaja sobre una versión completa del sistema, cada periodo se compone de cuatro fases, cada fase se subdivide en iteraciones y estas desarrollan en secuencia un conjunto de disciplinas. Cada una de estas disciplinas son un conjunto de actividades que son vinculadas con una parte del proyecto:

FASES, ITERACIONES Y DISCIPLINAS

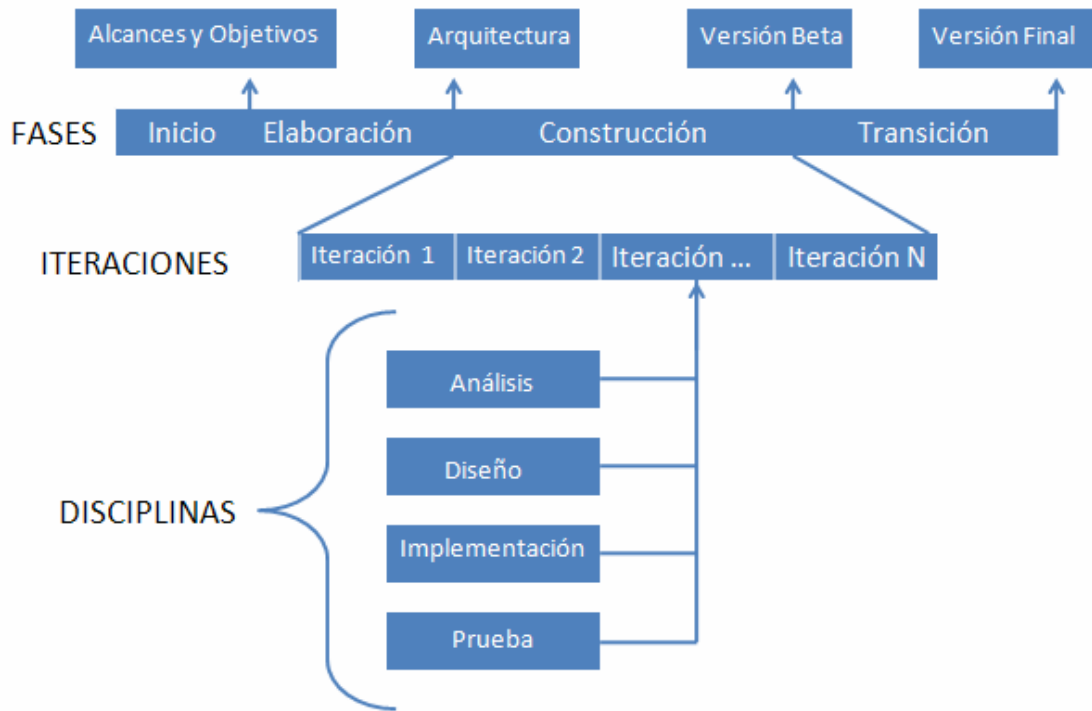


Figura 28. Fases, Iteraciones y Disciplinas de la Metodología RUP. Tomado de “Diseño de Sistemas”, Gustavo Torossi.

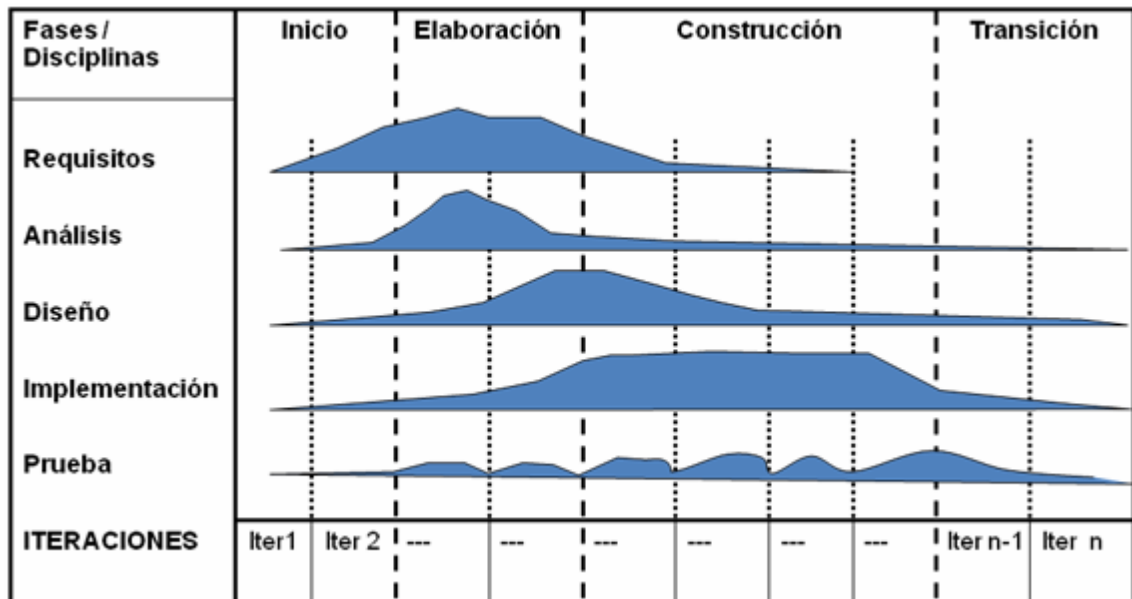


Figura 29. Tiempo de Ejecución de las Disciplinas en cada Fase. Tomado de “El Proceso Unificado de Desarrollo Software”, Addison Wesley.

FASE DE INICIO: en esta fase se define el ámbito y los objetivos del proyecto, se identifican las necesidades, los riesgos y las restricciones; se delimita el alcance y la viabilidad del resultado.

FASE DE ELABORACIÓN: los objetivos de esta fase son analizar el dominio del problema, establecer una base de la arquitectura software, desarrollar el plan de proyecto y eliminar los riesgos más importantes. En esta fase se debe disponer de un prototipo de la arquitectura software a partir del cual se pueda desarrollar el sistema.

“el termino arquitectura de un sistema consiste en la estructura o estructuras de las partes que lo componen (incluyendo el tiempo de diseño, el tiempo de pruebas y los componentes de software y hardware), en la naturaleza y propiedades relevantes de todas las partes (módulos con sus interfaces, las unidades de

hardware, objetos), y por último las relaciones e integridades entre todas las partes.”²⁰⁶

La base para el desarrollo y operación de un sistema es la arquitectura, que debe mostrar la estructura del sistema teniendo en cuenta un proceso de análisis de requisitos, diseño, implementación, y pruebas que puede realizarse en varias iteraciones.

Disponiendo la funcionalidad de UML se propone una arquitectura donde:

1. La etapa de análisis se llevara a cabo a través del desarrollo de los diagramas de casos de uso y diagrama de actividades.
2. La etapa de diseño contiene un diagrama que muestra subsistemas, interfaces que relacionan estos subsistemas y diagramas de clases.
3. En la etapa de implementación el trabajo se lleva a cabo bajo el seguimiento del diseño propuesto, el cual direccionará el cumplimiento de los requisitos del sistema.
4. La puesta en marcha se llevara a cabo con la utilización de los diagramas UML que servirán de soporte para el manual de usuario.

²⁰ D´Souza, F.F., Y Cameron A., Objects Components, and Frameworks with UML. The Catalisysma Approach, Addison-Weley, 1999.

De aquí en adelante el sistema se desarrolla de manera iterativa hasta obtener el resultado final.

FASE DE CONTRUCCIÓN: durante esta fase se desarrolla el sistema de forma iterativa hasta que esté listo para la puesta en marcha y a disposición de los usuarios. Como resultado de esta fase se obtiene el sistema con sus respectivos manuales.

En esta fase de construcción se espera:

1. Utilizar Lenguaje de desarrollo JAVA.
2. Implementar lógica para creación de modelos y parametrización.
3. Trabajar con archivos XML para la persistencia de los modelos (acceso y almacenamiento de datos).
4. Establecer algoritmos para manejo de variables lingüísticas y evaluación de reglas FUZZY.
5. Implementación de los algoritmos para defusificación y análisis de sensibilidad.
6. Implementación orientada a objetos.
7. Estricto desarrollo bajo los lineamientos del diseño propuesto, que direccionan el cumplimiento de los requisitos del sistema.

8. Generación y entrega del manual final.

FASE DE TRANSICIÓN: en esta fase se pone en operación y a disposición de los usuarios, cada una de las componentes del prototipo, teniendo en cuenta que pueden surgir cuestiones que requieran un desarrollo extra y que permitan dar un ajuste final a la herramienta.

En la fase de transición se espera enriquecer el sistema de acuerdo con los posibles aportes que se presenten, con el fin de mostrar un producto que se acomode a las condiciones de calidad que se requieren y a las expectativas que los usuarios tienen, además se incluirá el manual de usuario que presentará una guía sobre la utilización del prototipo.

2.3.2.2. DIMENSIÓN ESTÁTICA:

Representa el aspecto estático del proceso, las disciplinas, las actividades, los flujos de trabajo, los artefactos y las actividades.

PERFILES O PERFIL DE TRABAJO: se define el comportamiento y responsabilidad de las personas que trabajan en el desarrollo del proyecto; se estableció de la siguiente manera:

ROL	NOMBRE	COMPORTAMIENTO Y RESPONSABILIDAD
Director del proyecto	Fernando Ruiz	Coordina y participa en la realización de las actividades del proyecto, encargado de dar aval a las fases de desarrollo del mismo.

Codirector del proyecto	Fredy Estepa Martínez	Coordina y ejecuta acciones en pro del correcto desarrollo del proyecto en todas sus fases, y participa en la supervisión de las etapas de análisis, diseño, implementación y prueba.
Diseñador y Desarrollador del proyecto	Paola Andrea López Correa	Estudiante de Ingeniería de sistemas encargada del análisis, diseño, implementación y puesta en marcha de la herramienta, así como el desarrollo evolutivo de la misma.
Diseñador y Desarrollador del proyecto	Johanna Maritza Patiño Tejedor	Estudiante de Ingeniería de sistemas encargada del análisis, diseño, implementación y puesta en marcha de la herramienta, así como el desarrollo evolutivo de la misma.

Tabla 1. Descripción Funcional de los Involucrados en el Proyecto.

2.3.3. LENGUAJE DE MODELADO UNIFICADO

El Lenguaje de Modelamiento Unificado es un lenguaje grafico para especificar, visualizar, construir y documentar cada una de las partes que comprende el desarrollo software:

Especificar: establece la construcción de modelos precisos, no ambiguos y completos, cubriendo de esta manera las decisiones de análisis, diseño e implementación que se deben tener en cuenta al desarrollar.

Visualizar: interpretación de los símbolos utilizados en la notación UML.

Construir: es posible establecer correspondencia desde un modelo UML a un lenguaje de programación como C++ y Visual Basic, lo cual permite la generación de código a partir de un modelo UML en un lenguaje de programación y reconstruir un modelo UML a partir de una implementación. Es importante mencionar que en este proyecto no se utilizó ningún generador de código que interpretara los modelos UML y con base en este generará el código fuente de la aplicación.

Documentar: con UML se cubre la documentación de un sistema y sus detalles, además de proporcionar un lenguaje para expresar requisitos y pruebas, también ofrece un lenguaje de planificación de proyectos y gestión de versiones.

El lenguaje UML permite representar las fases de un proyecto comenzando desde el análisis con los casos de uso, el diseño con los diagramas de clase y la prueba con los diagramas de actividades.

2.3.4. DIAGRAMAS UML

Con el propósito de representar adecuadamente la arquitectura de un sistema, es necesario contar con varios diagramas o vistas, que constituyen una estructura de la arquitectura y muestran una parte del mismo como un conjunto de componentes que pueden relacionarse con otras estructuras para completar la visión integral del prototipo.

La arquitectura constituye un modelo del como está estructurado un sistema, permitiendo realizar los distintos análisis que orienten el proceso de toma de decisiones, por esta razón se hace necesario contar con una buena representación de la misma, en este punto UML permite mostrar la estructura del sistema, para ser utilizado en las etapas de desarrollo del mismo a través de diagramas que ilustran su funcionamiento.

Los elementos y las relaciones son agrupados en diagramas que representan diferentes aspectos del sistema y se presentan a continuación:

DIAGRAMA DE CLASES: Una clase es una categoría que posee atributos.

DIAGRAMA DE OBJETOS: Un objeto es una instancia de clase o categoría, estos poseen atributos o propiedades y acciones, estas acciones son las actividades que el objeto es capaz de realizar, por esto los diagramas de objetos representan instantáneas de instancias de los elementos encontrados en los diagramas de clase.

DIAGRAMAS DE CASOS DE USO: Un diagrama de casos de uso es una descripción de las acciones de un sistema desde el punto de vista de un usuario, ayuda a comprender la forma en que un sistema debe comportarse y ayuda a obtener los requerimientos desde el punto de vista del usuario.

DIAGRAMA DE SECUENCIAS Y DIAGRAMA DE COLABORACIÓN: Son un tipo de diagrama de interacción que consta de un conjunto de objetos y sus relaciones e incluyen los mensajes que pueden ir entre ellos. Un diagrama de secuencia es un diagrama de interacción que resalta la ordenación temporal de los mensajes, un diagrama de colaboración es un diagrama de interacción que

resalta la organización estructural de los objetos que envían y reciben mensajes.

DIAGRAMA DE ESTADOS: representa los estados en los que se puede encontrar un objeto junto con las transiciones que ocurren entre los estados, muestra el punto inicial y final de una secuencia de cambios de estado y muestra las condiciones de un solo objeto.

DIAGRAMA DE ACTIVIDADES: Es un tipo especial de diagrama de estados que se ha diseñado para mostrar una visión simplificada de lo que ocurre en un proceso (flujo de actividades dentro de un sistema). Son importantes al modelar el funcionamiento de un sistema y hacen notar el flujo de control entre objetos.

DIAGRAMA DE COMPONENTES: muestra la organización y las dependencias entre un conjunto de componentes, los componentes se corresponden con una o más clases, interfaces o colaboraciones. Los tipos de componentes son: componentes de distribución (conforman el fundamento de los sistemas ejecutables), componentes para trabajar el producto (a partir de los cuales se ha creado los componentes de distribución), componentes de ejecución (creados como resultado de un sistema en ejecución).

DIAGRAMA DE DESPLIEGUE: estos diagramas muestran la configuración de nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos.

El presente documento profundiza los diagramas de casos de uso, diagramas de clases y diagramas de actividades ya que son los más representativos en la realización del proyecto.

2.4. TECNOLOGÍA DE DESARROLLO UTILIZADA

2.4.1. JAVA

Java es un lenguaje orientado a objetos (OO) que permite el diseño software de tal manera que los diferentes tipos de datos²¹ que utilice estén unidos a sus operaciones, también ofrece la característica de tener independencia de la plataforma, es decir que los programas escritos en lenguaje Java pueden ejecutarse en cualquier tipo de Hardware y además posee las características de encapsulación, herencia y polimorfismo.

La utilización de este lenguaje ofrece una serie de ventajas y características entre las que se encuentran:

VENTAJAS

- La realización de verificaciones en busca de problemas que puedan surgir en tiempo de compilación y/o tiempo de ejecución.
- La comprobación de tipos para detectar errores en el menor tiempo posible en el ciclo de desarrollo.
- Declaración de métodos para reducir la posibilidad de error.

²¹ Java trabaja con sus datos como objetos y con interfaces a esos objetos.

- Implementación de arrays con comprobación de límites, para evitar la posibilidad de sobrescribir o dañar la memoria a través de punteros que señalan zonas equivocadas.
- Comprobación de los bytes-codes resultado de la compilación de un programa Java.

CARACTERÍSTICAS

- *PORTABLE*: Java implementa estándares de portabilidad con el fin de facilitar su desarrollo y construye sus interfaces de usuario a través de un sistema abstracto de ventanas que pueden ser implementadas en distintos entornos.
- *INTERPRETADO*: El intérprete Java ejecuta el código objeto y enlaza programas lo cual resulta beneficioso ya que consume menos recursos que al compilar.
- *MULTITHREADED*: Permite actividades simultaneas en un programa, mejor rendimiento interactivo y mejor comportamiento en tiempo real.

Para el desarrollo de programas en Java es necesaria la utilización de dos componentes en el sistema en donde se va a ejecutar: la maquina virtual y el conjunto de librerías que proporciona los servicios que requiera la aplicación:

MÁQUINA VIRTUAL DE JAVA: la función de este componente es ejecutar el código que resulta de la compilación del código fuente, (generalmente

conocido como bytecode²²) lo cual permite que una aplicación Java sea ejecutada en sistemas con arquitecturas distintas.

LIBRERÍAS DE JAVA: constituyen un conjunto de funciones para realizar tareas como por ejemplo el manejo de listas de elementos y operaciones en cadenas de caracteres, además proporcionan una interfaz para tareas que dependen del hardware, de la plataforma de destino y del sistema operativo.

Se decidió utilizar el lenguaje de programación Java, ya que los programas que se crean en ella se pueden ejecutar sin cambios en los diferentes tipos de arquitecturas y plataformas, posee su propia maquina virtual que propicia la portabilidad en ejecución y además permite referenciar el patrón MVC (Modelo vista controlador) del cual se tratara con detalle más adelante. Por último cabe resaltar que al ser software libre, constituye una herramienta de gran utilidad para el desarrollo de aplicaciones.

2.4.2. ARCHIVOS XML

La forma como se almacenan los datos se denomina persistencia de datos, en Java uno de los mecanismos más utilizados es la serialización y se define como:

“un proceso por el que un objeto cualquiera se puede convertir en una secuencia de bytes con la que más tarde se podrá reconstruir el valor de sus variables. Esto permite guardar un objeto en un archivo o mandarlo por la red.”²³

²² Código intermedio que contiene un programa ejecutable cuyo contenido es el código de la compilación del código fuente.

²³ García 1999

XML funciona como un lenguaje que hace referencia a otros lenguajes y utiliza etiquetas (las etiquetas describen parte de un archivo) que marcan los datos de manera que se puedan almacenar en un archivo XML y se puedan gestionar adecuadamente cuando se exporten a otros archivos; estas etiquetas XML no contienen información acerca de cómo se deben mostrar los datos ni sobre el formato que deben tener sino que sirven estrictamente para identificar contenido, XML funciona como un estándar para el intercambio de información entre diferentes plataformas y permitirá la compatibilidad de la serialización sin importar la versión de Java que se esté utilizando.

XML se encuentra en registros estructurados, aplicaciones web, registros de datos, presentaciones gráficas como interfaces de usuario, entre otros; además presenta ventajas tales como:

Los formatos XML se basan en texto, lo que los hace más legibles, fáciles de documentar y en ciertas ocasiones fáciles de depurar.

Los documentos XML utilizan infraestructura para HTML, lo cual permite que se transfiera XML a través de servidores de seguridad.

El análisis XML está bien definido e implementado, lo que posibilita recuperación de información de documentos XML en diversos entornos.

Es importante resaltar que los archivos XML requieren mayor espacio de almacenamiento y más tiempo de procesador para la compresión, además el análisis XML puede ser más lento que el análisis de los formatos binarios y puede requerir más memoria.

La arquitectura de api java para uniones XML (JABX) proporciona una manera rápida de crear uniones entre documentos XML y los objetos de Java, para un esquema que especifica la estructura de datos XML, el compilador Jabx genera un conjunto de clases de Java que contiene el código para analizar los documentos XML compilados en bytecodes, un formato intermedio diseñado para transportar el código a múltiples plataformas hardware y software y de esa manera cualquier sistema con una máquina virtual Java puede interpretar los bytecodes, aunque esto puede representar una pequeña desventaja donde Java no es un lenguaje tan rápido como otros pero se prefirió la versatilidad del prototipo que su velocidad, en cualquier caso su rapidez no se disminuye.

Los beneficios de utilizar esta arquitectura son:

Uso de tecnología Java y XML.

Garantiza datos validos.

Es rápida y fácil de usar.

Puede restringir datos.

Es personalizable y extensible.

La clase *java.beans.XMLEncoder* permite la persistencia de objetos como documentos XML, esta clase convierte el objeto y cada uno de sus datos a un documento XML. La utilización de archivos XML, por un lado permite realizar cambios en las versiones ya que los objetos que son persistentes no poseen dependencias y pueden ser intercambiados entre entornos que poseen versiones

diferentes de algunas clases o maquinas virtuales y por otro lado es tolerante a fallos ya que permite reportar errores causados por daños de archivos o cambios de API sin afectar la carga de partes del documento que no sean afectadas por este.

2.4.3. ANÁLISIS DE SENSIBILIDAD DE MODELOS

Hoy en día muchos cambios técnicos y tecnológicos, a nivel teórico y experimental, involucran formas de optimización para encontrar soluciones que mejoren los resultados arrojados y permitan de manera aproximada establecer aquel que mejor se acople a las variaciones que se realicen en sus variables de entrada, a continuación se presenta el concepto de algunos métodos comúnmente utilizados para realizar análisis de sensibilidad:

Análisis Post-Optimal: consiste en encontrar una nueva solución óptima a partir de una solución actual que está determinada por ciertos parámetros.

Algoritmo de Ramificación y Acotamiento: este algoritmo no considera restricciones de integralidad y construye un árbol con soluciones enteras, particionando el conjunto de soluciones factibles de modo de descartar soluciones fraccionarias.

Máximos y Mínimos: es el problema clásico de programación lineal en el cual se encuentran los valores máximos o mínimos de una función lineal bajo restricciones representadas por ecuaciones o desigualdades.

El análisis de sensibilidad seleccionado para este proyecto se llevara a cabo mediante la implementación de algoritmos que se ajustaran debidamente al

software y que darán como resultado un producto que se acople a las necesidades de los usuarios, las cuales serán detectadas en la fase de análisis.

2.4.4. PATRÓN DE DESARROLLO DE SOFTWARE

El patrón modelo vista controlador MVC es un patrón encargado de separar los datos de una aplicación, la interfaz grafica y la lógica de control con el objetivo de reutilizar componentes fácilmente, hacer modificaciones en código que ya ha sido realizado, mejorar el aspecto de un programa o mejorar un algoritmo, en conclusión facilitar la separación de código para que se lleven a cabo tareas independientes, sin que se presenten alteraciones y/o interrupciones en las demás partes al momento de realizar compilaciones.

MODELO: Es la representación de la información con la que el sistema opera.

VISTA: presenta el modelo en una interfaz grafica que sea agradable al usuario, generalmente denominada interfaz de usuario.

CONTROLADOR: Responde a los eventos que realiza el usuario y llama a los cambios en el modelo y algunas veces en la vista. En las aplicaciones el controlador recibe la petición que hace el usuario, interactúa con el modelo y dispone de los datos a la vista.

Generalmente el flujo que sigue el control es el siguiente:

El usuario interactúa con la interfaz de usuario, el controlador recibe la notificación de la acción, gestiona el evento, accede al modelo actualizándolo y si es necesario modificándolo de acuerdo a la acción solicitada, además delega a los objetos de la

vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para poder generar la interfaz apropiada para el usuario donde se reflejan los cambios solicitados. El modelo no tiene conocimiento directo sobre la vista, sin embargo existe un patrón observador que se utiliza para proveer una comunicación indirecta entre los dos, permitiendo al modelo notificar los cambios, la interfaz de usuario espera nuevas interacciones del usuario, comenzando nuevamente con el ciclo.

CAPÍTULO III

3. MARCO METODOLÓGICO

Como se describió anteriormente el desarrollo del proyecto se lleva a cabo utilizando el proceso unificado de desarrollo de software, teniendo en cuenta cada una de las fases y las disciplinas que sobre estas incurren.

3.1. FASE DE INICIO

Teniendo en cuenta que hoy en día existen diferentes tipos de herramientas que ayudan a realizar y solucionar operaciones matemáticas, aritméticas entre otras, y que permiten la representación de datos, la implementación de algoritmos, la creación de interfaces de usuario (GUI) entre otras, cabe la necesidad de desarrollar una herramienta que además de realizar estas funciones permita la utilización de lógica difusa para la resolver inconvenientes y dar soluciones con un grado de estabilidad acorde a determinadas necesidades.

Se determina establecer las herramientas necesarias para realizar cada una de las fases especificadas sobre las que se construye el prototipo, teniendo en cuenta el correcto diseño de las funcionalidades del mismo como una parte fundamental para cumplir las expectativas planteadas al iniciar la ejecución del proyecto.

3.1.1. PLANEACIÓN DE LA FASE:

Se planteo realizar las siguientes actividades para determinar el contenido que abarca el desarrollo de la herramienta:

Recopilación de información bibliográfica necesaria para identificar las características de la herramienta que se va a desarrollar.

Elaboración del marco teórico con la información recopilada.

Selección de los casos de uso funcionales para la herramienta.

Generalmente una herramienta es utilizada por diferentes tipos de usuario, cada tipo de usuario se representa por un actor y estos utilizan la herramienta interactuando con los casos de uso. Un caso de uso representa una secuencia de acciones que se llevan a cabo para llegar a un resultado de valor para un actor.

3.1.2. ANÁLISIS DEL ALCANCE

Durante esta fase se definió el alcance del sistema propuesto y se plantearon los objetivos del mismo que optimizaran los requisitos presentados. A continuación se muestra una breve explicación de las actividades que se llevan a cabo para la realización del prototipo, con el fin de presentar una visión general del desarrollo.

Se realiza un estudio preliminar del concepto de lógica difusa, teniendo en cuenta aspectos como funciones de pertenencia, variables lingüísticas, conjuntos difusos y operaciones entre conjuntos difusos; además se realiza un estudio referente a sistemas difusos de dos capas y algunas aplicaciones de estos sistemas, que son necesarios para llevar a cabo el desarrollo del prototipo de herramienta, objetivo central del presente proyecto.

En la identificación de los requerimientos de información se tiene en cuenta el software matemático Matlab que permite ejecutar la herramienta de lógica difusa,

para determinar posibles diferencias y como base en la ejecución del proyecto, a fin de presentar algunas modificaciones tales como puntos máximos y mínimos, representación de modelos de dos capas y análisis de sensibilidad a dichos modelos, correspondiendo estos dos últimos con los objetivos específicos planteados.

Teniendo en cuenta el diseño, se presenta una estructura organizada, versátil y fácil de manejar que cumpla con las expectativas de desarrollo y que muestre gráficamente y de manera satisfactoria los resultados arrojados en la representación de modelos, que sean acordes con las necesidades y requerimientos que se presentan durante el análisis.

Modelo de procedimientos: se realizan mejoras en los procedimientos estudiados en la etapa de análisis para lograr una mejor efectividad en las operaciones, aprovechando las ventajas que la herramienta pueda brindar.

Diseño de la interfaz de usuario: la interfaz de usuario se diseña de manera que su entorno sea amigable, además se traducen los requisitos en una representación de la herramienta para obtener la mejor calidad.

En la implementación, se considera el desarrollo de la herramienta de acuerdo a lo especificado con anterioridad.

Construcción de los parámetros base de la herramienta: En este punto se lleva a cabo la construcción básica de los parámetros que harán posible realizar los aspectos mencionados en la etapa de análisis, dentro de la implementación se tendrán en cuenta los mecanismos de parametrización, implementación del modelo vista controlador, manipulación de variables

lingüísticas y funciones de pertenencia, evaluación de reglas y algoritmos de defusificación y de análisis de sensibilidad así como la implementación de la persistencia de datos, cada uno de estos factores se profundizan en la fase de construcción.

Construcción del ambiente de desarrollo: se desarrolla toda la interfaz grafica del prototipo, durante esta etapa se realiza la codificación de los programas que permiten la integración del usuario final y el software a desarrollar, organizado de acuerdo a la estructura de menús e interfaces que se manejan, de manera que muestra las opciones que faciliten la navegación de los usuarios a través de la herramienta.

Se elaboran documentos soportes para el correcto funcionamiento y además el manual de usuario con una descripción esquemática y gráfica del prototipo de la herramienta computacional que facilite su uso.

Por último, en el momento en que se obtiene la aplicación se realizan pruebas con datos que son extraídos de artículos publicados, realizando la comprobación de su eficacia a través de una comparación de resultados con Matlab, esto para analizar el funcionamiento y la lógica que posee la herramienta y la viabilidad que representa el desarrollo de la misma para la ejecución de actividades que incluyan lógica difusa.

3.1.3. DESCRIPCIÓN DE NECESIDADES:

A continuación se presenta una breve descripción de las principales necesidades:

ID	Nombre	Descripción	Prioridad
1	Presentar un Prototipo de herramienta computacional que utilice lógica difusa.	Se proporciona un prototipo de herramienta computacional que permita la inclusión de variables lingüísticas, Reglas de inferencia y conjuntos borrosos para la realización de modelos difusos de dos capas.	Alta
2	Diseñar un ambiente que sea agradable al usuario.	Se diseña un ambiente en el cual el usuario pueda interactuar con la herramienta de forma clara y comprensible en la realización de modelos.	Alta
3	Modificación de los datos de la herramienta.	Presentar la posibilidad de crear nuevas variables, cambiar parámetros de los conjuntos borrosos y observar gráficamente dichos conjuntos, para cada una de las variables lingüísticas, entre otras cosas.	Alta
4	Presentar escenarios de estabilidad	Presentar escenarios de estabilidad donde se puedan observar los cambios que presentan las variables de salida frente a modificaciones realizadas en las variables de entrada de un	Alta

		determinado modelo.	
5	Presentar manual de usuario.	Se elabora un manual de usuario que representa una guía en el uso de la herramienta, de manera que se facilite el reconocimiento de cada una de sus componentes.	Media

Tabla 2. Descripción de las Principales Necesidades del Proyecto.

3.1.4. FUNCIONES DEL PRODUCTO

Representación de modelos difusos de una y dos capas, donde se puede visualizar gráficamente la estructura y características que poseen.

Crear, actualizar y eliminar variables lingüísticas y conjuntos borrosos permitiendo a estos últimos la asignación de parámetros correspondientes a las funciones de pertenencia establecidas.

Creación de reglas de inferencia difusa para cada una de las capas que intervienen en el modelo.

Representación grafica de los resultados obtenidos para cada una de las capas del modelo.

Representación de escenarios de estabilidad y puntos máximos y mínimos.

3.1.5. RIESGOS DEL PROYECTO:

Todos los proyectos informáticos tienen riesgos que son inherentes a la naturaleza de la actividad, entre más pronto sean identificados en el ciclo de vida del proyecto, más factible será la elaboración de un plan de mitigación y contingencia y con seguridad será más probable la elaboración de trabajos acertados. La mayor parte de riesgos que se encuentran en un proyecto software no se descubren sino hasta la etapa de implementación, siendo la corrección de estos un gasto no previsto, que incluso pueden conllevar a la cancelación del proyecto:

DESCRIPCIÓN	PRIORIDAD	IMPACTO	MONITOR	CONTINGENCIA
Errores en la escritura de la información requerida para la ejecución de un determinado modelo.	Critico	Se afecta el correcto funcionamiento de la herramienta debido a que la información registrada es incorrecta.	Usuario de la herramienta.	Validación de los campos de registro, con el fin de corroborar que el tipo de dato sea el correcto para cada requerimiento.

Omisión de herramientas de desarrollo	Critico	Se paraliza la ejecución del proyecto	Desarrollado	En el momento en que se inicia el proyecto contar con las herramientas necesarias para su ejecución.
---------------------------------------	---------	---------------------------------------	--------------	--

Tabla 3. Riesgos del Proyecto.

3.2. FASE DE ELABORACIÓN

En esta etapa se brinda al equipo de desarrollo las herramientas necesarias para hacer del análisis y del diseño las bases sobre las cuales se da inicio a la elaboración de la herramienta, haciendo énfasis en el diseño de las funcionalidades, como parte indispensable para cumplir las expectativas de los usuarios en torno a la aplicación y teniendo en cuenta las directrices mediante las cuales se regirá el desarrollo de la misma.

3.2.1. PLANEACIÓN DE LA FASE:

La misión de la fase de elaboración dentro de la metodología elegida es lograr definir la arquitectura del prototipo, con el propósito de cumplir esto, se trazan algunas actividades a realizar.

Constituir una arquitectura concreta para el desarrollo del prototipo que permita guiar la fase de construcción con la seguridad que se abarcarán todos los

requisitos propuestos los cuales se exponen más adelante dentro de ésta misma fase.

Definir el lenguaje de programación y los recursos idóneos a la naturaleza del proyecto que aseguren el correcto desarrollo del mismo.

3.2.2. DEFINICIÓN DE REQUISITOS

Establecer los requisitos que servirán como base en el desarrollo del prototipo y sobre los cuales se comenzara la ejecución del mismo.

3.2.2.1. REQUISITOS DEL SISTEMA

Para los requisitos se tuvieron en cuenta los componentes en la construcción de un sistema difuso, igualmente se contempló la necesidad de ofrecer un entorno en el cual se pueda observar la estabilidad y el comportamiento de un determinado modelo. Se considera que estos requerimientos son de gran importancia para el desarrollo del proyecto ya que en fases posteriores del mismo se podrá verificar el cumplimiento de las tareas propuestas para el prototipo. De manera que estos requisitos queden plasmados de forma más explícita se realizó la tabla que aparece a continuación, en la que se incluye tanto el nombre del requisito como su respectiva descripción.

NOMBRE	DESCRIPCIÓN
Crear Modelos Difusos	Crear modelos difusos ya sea de una o dos capas con variables lingüísticas, conjuntos borrosos y

	reglas de inferencia difusa.
Abrir Modelos	Permitir que se puedan abrir modelos que fueron guardados anteriormente.
Guardar Modelos	Permitir que se puedan guardar los modelos construidos.
Crear y Modificar Variables Lingüísticas	Creación de múltiples variables lingüísticas según requiera el usuario, donde se pueda especificar si dicha variable es de entrada o salida, si pertenece a la primera capa o a la segunda; donde también se pueda asignar a la variable nombre, unidades, valor mínimo y máximo y vincular a ella sus respectivos conjuntos borrosos, además realizar modificaciones y eliminar variables.
Crear y Modificar Conjuntos Borrosos	Crear conjuntos borrosos que pertenezcan a determinadas variables lingüísticas y a los cuales se les permita asignarle nombre, función de pertenencia y color de línea de cada conjunto, además,

	ofrecer un espacio para incluir los parámetros de las funciones de pertenencia y dar la opción de hacer modificaciones y eliminar conjuntos borrosos.
Crear y Modificar Reglas Difusas	Crear las reglas de inferencia difusa que el usuario considere pertinentes tanto para la primera capa como para la segunda, dar la posibilidad de escoger si el conector de determinada regla es “y” u “o” y permitir que se puedan agregar y eliminar reglas.
Visualizar Resultados	Observar los resultados del proceso de defusificación al mostrar las reglas de inferencia de forma gráfica, permitir modificar el valor de cualquiera de las variables de entrada para visualizar su efecto en las variables de salida, dar la posibilidad de distinguir los resultados de las variables de salida de forma gráfica y numérica.
Visualizar Escenarios de Estabilidad	Visualizar los gráficos de las situaciones donde se presente la mayor inestabilidad junto con datos

	que soporten esas situaciones de manera que se pueda reconocer los puntos en los que las variables de salida presentan cambios; dar la posibilidad de que el usuario escoja en cuantos puntos desea que se haga este procedimiento.
Visualizar Máximos y Mínimos	Permitir la observación de los datos correspondientes a las variables de entrada y salida de las situaciones en las que se presentan los picos más altos y los mayores descensos para las variables de salida, además, dar la opción que el usuario escoja alguna de las variables de salida para ver sus máximos y mínimos.

Tabla 4. Requisitos de la Herramienta.

3.2.2.2. REQUISITOS ADICIONALES

Los requisitos de plataforma hardware son los siguientes:

1. Equipo de Desarrollo

Procesador superior a 2.0 GHz.

Memoria RAM mayor o igual a 512 MB.

Disco duro de 40 GB en adelante.

2. Usuario

Procesador superior a 1.6 GHz.

Memoria RAM mayor o igual a 512 MB.

Los requisitos para la plataforma software son los siguientes:

Máquina Virtual de Java.

3.2.3. ARQUITECTURA SOFTWARE

El lenguaje de programación escogido para el desarrollo del prototipo fue JAVA debido a que es un lenguaje de programación orientado a objetos que permite que el código tenga una mejor organización y desempeño, gracias a esto, el código se puede agrupar en paquetes facilitando así el manejo del patrón de arquitectura modelo vista controlador (MVC) que también fue adoptado para el desarrollo, resaltando que con su utilización, los datos, la interfaz gráfica y la lógica de control pueden ser separados en tres componentes distintos según los cuales cada uno consigue manejar independientemente sus respectivas funciones.

Se utilizó el Api jaxb para entrelazar documentos XML con objetos JAVA. Adicionalmente, se tuvo en cuenta que el prototipo debe poder usarse bajo distintas arquitecturas y con diversos sistemas operativos, motivo por el que se pensó en JAVA debido a su indiferencia a la arquitectura y su portabilidad, gracias al uso de los bytecodes en JAVA.

Existe un api de Java que ofrece una biblioteca de clase llamada Jfreechart que se encarga de la representación de gráficas, muy útil en este caso teniendo en cuenta que para el prototipo los gráficos son de gran importancia.

Para la persistencia de datos, tomando en cuenta el lenguaje de programación escogido, se decidió utilizar la tecnología XML puesto que Java posee recursos especialmente diseñados para interactuar con ésta tecnología.

3.2.3.1. DIAGRAMAS CASOS DE USO, CLASES Y ACTIVIDADES

A continuación se presentan los diagramas que se utilizaron para representar la estructura de la arquitectura de la herramienta:

DIAGRAMA CASOS DE USO:

Diagrama caso de uso Usuario

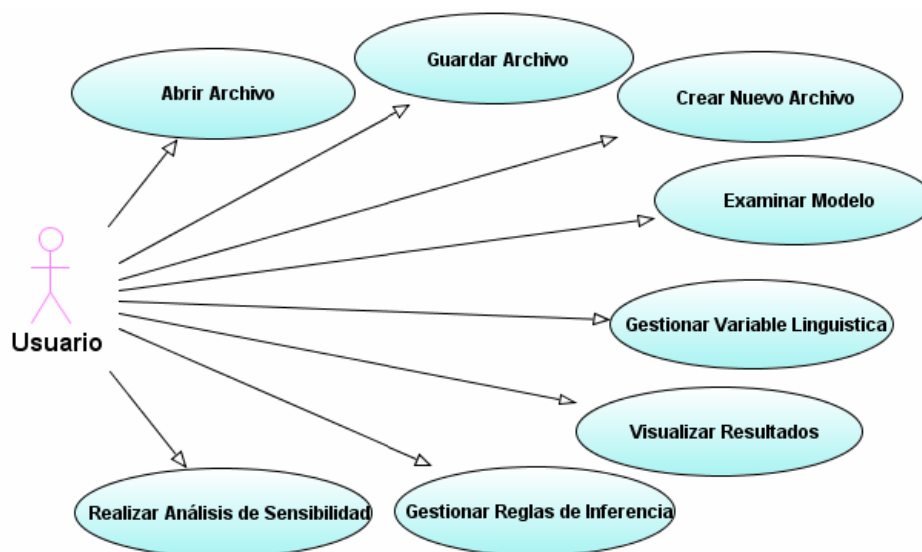


Figura 30. Diagrama Caso de Uso Usuario.

Id del Caso de Uso	001
Resumen:	Funciones de la Aplicación.
Actores Directos:	Usuario.
Entradas:	Información requerida para ejecutar aplicación.
Salidas:	Representación de modelos de una y dos capas, permitiendo hacer análisis de sensibilidad y conociendo la susceptibilidad de los resultados arrojados con respecto a las decisiones de diseño que se toman.
Precondiciones:	Se debe realizar una lectura previa al manual de usuario, esto con el fin de tener claridad en cuanto al manejo del prototipo.
Escenario Principal de Éxito:	Manejo adecuado de la herramienta, a partir de la utilización del manual de usuario.
Postcondiciones	Presentación de una visión general del funcionamiento de la herramienta, sus características y la forma como operan cada una de sus componentes.

Id del Caso de Uso	001
Escenario de Extensiones Alternativas:	<p>A partir de este caso de uso se da inicio al estudio de cada una de las ventanas que comprende el prototipo.</p> <ol style="list-style-type: none"> 1. Esquema. 2. Variable Lingüística. 3. Reglas Capa 1. 4. Reglas Capa 2. 5. Resultados Capa 1. 6. Resultados Capa 2. 7. Estabilidad. 8. Máximos y Mínimos.
Criterios de aceptación	<p>Permite al usuario interactuar con cada una de las ventanas que posee la herramienta para la elaboración de modelos difusos de una y dos capas.</p>

Tabla 5. Descripción Caso de Uso Usuario.

Diagrama Caso de Uso Examinar Modelo

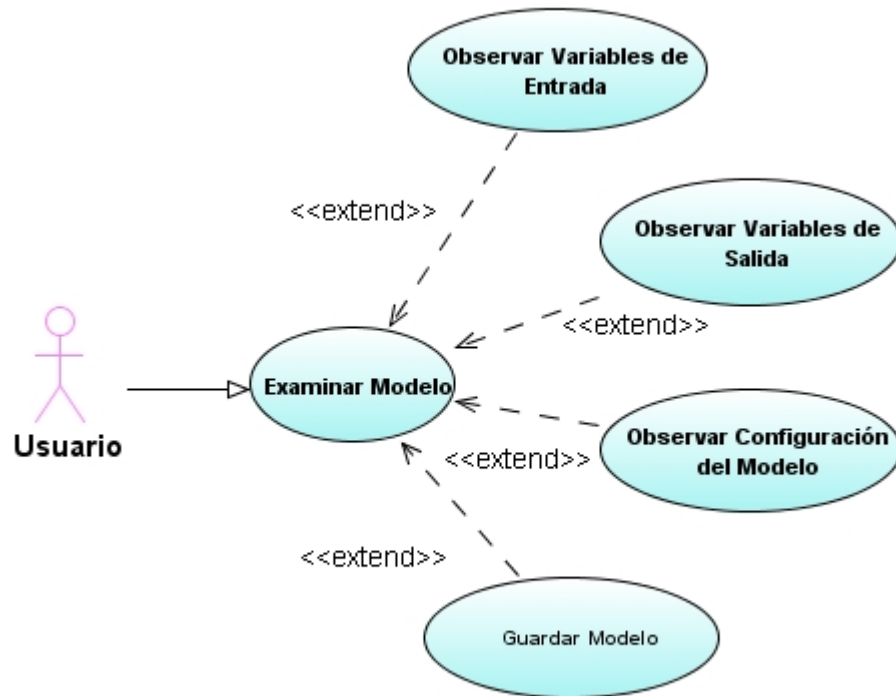


Figura 31. Diagrama Caso de Uso Examinar Modelo.

Id del Caso de Uso	002
Resumen:	Visualización de la configuración general del modelo.
Actores Directos:	Usuario.

Id del Caso de Uso	002
Entradas:	<ol style="list-style-type: none"> 1. Información requerida para visualizar la configuración del modelo. 2. Nombre del modelo.
Salidas:	<ol style="list-style-type: none"> 1. Variables de entrada. 2. Variables de salida.
Precondiciones:	<ol style="list-style-type: none"> 1. Ingresar los datos requeridos para representar las variables de entrada y las variables de salida. 2. Asignar el nombre con el que se guarda el modelo.
Escenario Principal de Éxito:	Introducir los criterios necesarios para la representación del modelo.
Postcondiciones	Se visualiza el nombre que se le asigno al modelo así como cada una de las variables de entrada que se incluyen en el proceso y las variables de salidas que surgen del mismo.
Requerimientos Especiales	En el momento en que se disponga a guardar el modelo especificar la ruta de almacenamiento y el tipo de archivo al que corresponde (XML).
Criterios de aceptación	Dar al usuario información general sobre las variables, nombre del modelo y método de defusificación utilizado.

Tabla 6. Descripción Caso de Uso examinar Modelo.

Diagrama Caso de Uso Crear Variable Lingüística

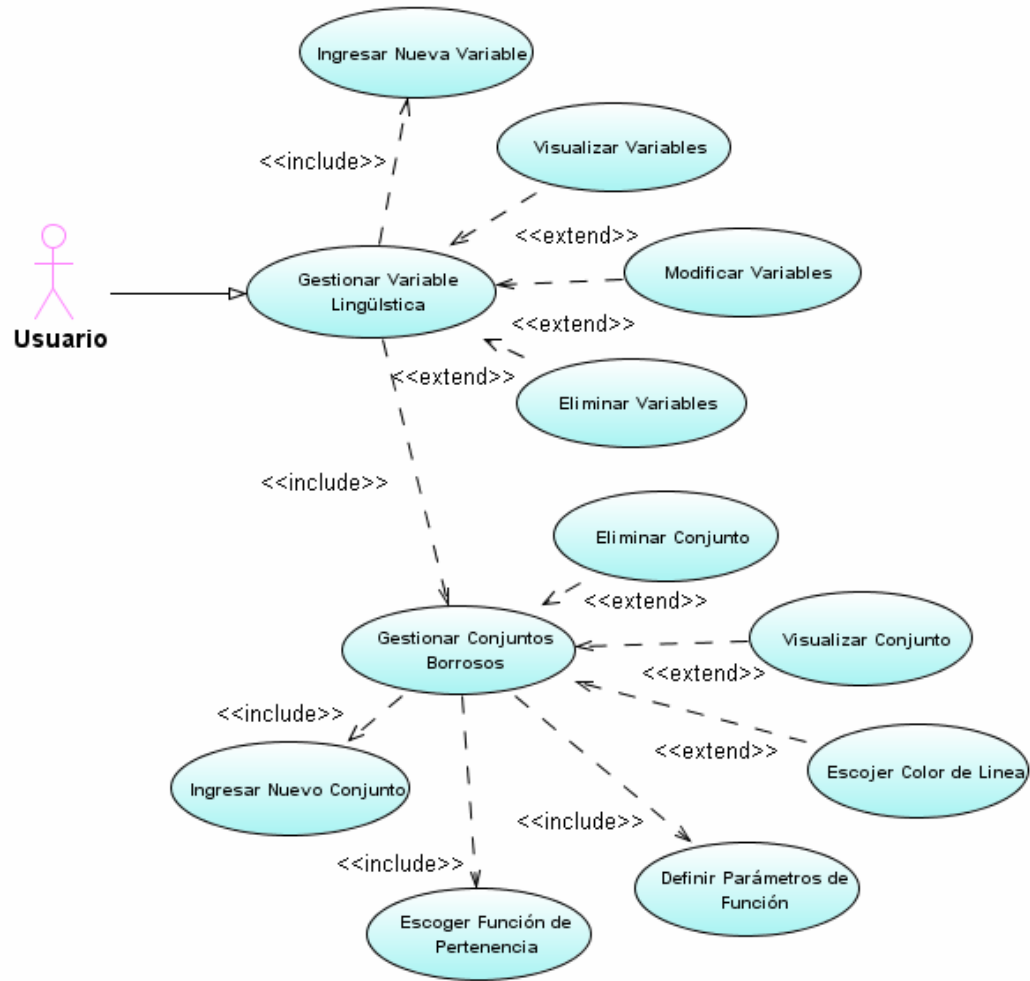


Figura 32. Diagrama Caso de Uso Gestionar Variable Lingüística.

Id del Caso de Uso	003
Resumen:	Gestionar la creación de las variables lingüísticas y los conjuntos borrosos.
Actores Directos:	Usuario.
Entradas:	

Id del Caso de Uso	003
	<ol style="list-style-type: none"> 1. Variable de entrada Capa 1. 2. Variable de entrada Capa 2. 3. Variable de salida Capa 1. 4. Variable de salida Capa 2. 5. Unidades de la variable. 6. Valor máximo y mínimo de las variables. 7. Nombre completo de las variables. 8. Nombre del conjunto borroso. 9. Tipo de función de pertenencia. 10. Color de línea del conjunto borroso. 11. Parámetros de la función de pertenencia.
Salidas:	<ol style="list-style-type: none"> 1. Representación grafica de la Variable Lingüística Parametrizada. 2. Representación grafica del Conjunto Borroso.
Precondiciones:	<ol style="list-style-type: none"> 1. Es necesaria la creación de un rango de valores para la variable lingüística. 2. Los conjuntos borrosos deben estar relacionados con algunas de las funciones de pertenencia que en la herramienta se ofrecen teniendo en cuenta los parámetros que para cada una de estas se requieren.
Escenario Principal de Éxito:	<ol style="list-style-type: none"> 1. Ingreso de datos. 2. Del listado de funciones de pertenencia, seleccionar la que corresponda con el interés del usuario.

Id del Caso de Uso	003
Postcondiciones	Se visualizan las diferentes variables y conjuntos que se crearon para utilizarlos en los resultados de cada una de las capas y en el proceso de estabilidad y localización de puntos máximos y mínimos
Escenario de Extensiones Alternativas:	<ol style="list-style-type: none"> 1. Reglas Capa 1. 2. Reglas Capa 2. 3. Resultados Capa 1. 4. Resultados Capa 2. 5. Estabilidad. 6. Máximos y Mínimos.
Requerimientos especiales	Para la creación de los conjuntos borrosos es indispensable la existencia de una variable lingüística a la cual este pueda pertenecer.
Criterios de aceptación	<ol style="list-style-type: none"> 1. Permite al usuario la creación de variables lingüísticas y conjuntos borrosos. 2. Permite interactuar directamente con el escenario en donde se muestran las variables y conjuntos.

Tabla 7. Descripción Caso de Uso Gestionar Variable Lingüística.

Diagrama Caso de Uso Gestionar Reglas de Inferencia

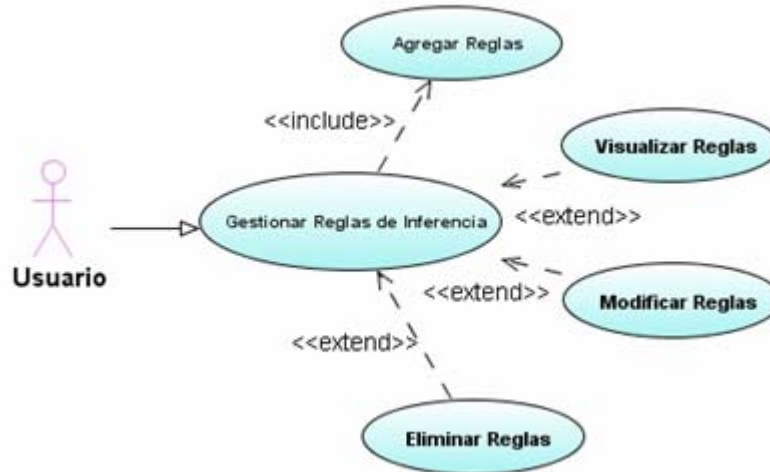


Figura 33. Diagrama Caso de Uso Gestionar Reglas de Inferencia.

Id del Caso de Uso	004
Resumen:	Realizar acciones referentes a las reglas.
Actores Directos:	Usuario.
Entradas:	1. Variables lingüísticas.
Salidas:	1. Reglas de Inferencia capa 1. 2. Reglas de Inferencia capa 2.
Precondiciones:	Variables de entrada y salida.
Escenario Principal de Éxito:	Se incluyen solamente las reglas relevantes para el modelo.
Postcondiciones	Observación de resultados y análisis de sensibilidad.

Id del Caso de Uso	004
Escenario de Extensiones Alternativas:	<ol style="list-style-type: none"> 1. Esquema general del modelo. 2. Variables Lingüísticas.
Requerimientos especiales	Seleccionar conector “Y” u “O” para la creación de reglas.
Criterios de aceptación	Permite al usuario crear, modificar o eliminar reglas según una combinación de variables de entrada y salida.

Tabla 8. Descripción Caso de Uso Gestionar Regla de Inferencia.

Diagrama Caso de Uso Visualizar Resultados

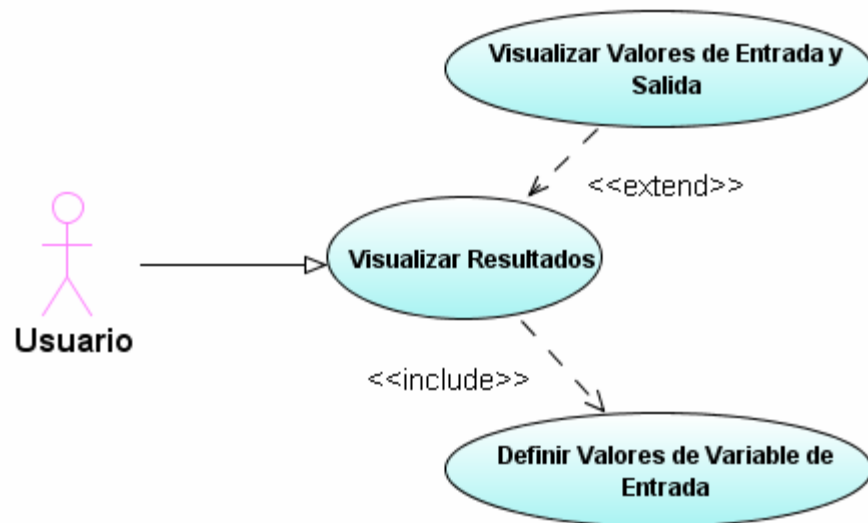


Figura 34. Diagrama Caso de Uso Visualizar Resultados.

Id del Caso de Uso	005
Resumen:	Conocer resultados
Actores Directos:	Usuario
Entradas:	Reglas de inferencia.
Salidas:	Valores de variables de salida.
Precondiciones:	Reglas de inferencia.
Escenario Principal de Éxito:	Ingreso de valores de entrada representativos y relevantes para el modelo.
Postcondiciones	Análisis de sensibilidad.
Escenario de Extensiones Alternativas:	<ol style="list-style-type: none"> 1. Esquema general del modelo. 2. Variables Lingüísticas. 3. Reglas de capa 1 y 2.
Requerimientos especiales	Fijar valores de entrada para todas las variables de entrada.
Criterios de aceptación	Permite al usuario observar los resultados de las variables de salida.

Tabla 9. Descripción Caso de Uso Visualizar Resultados.

Diagrama de Casos de uso Realizar Análisis de sensibilidad

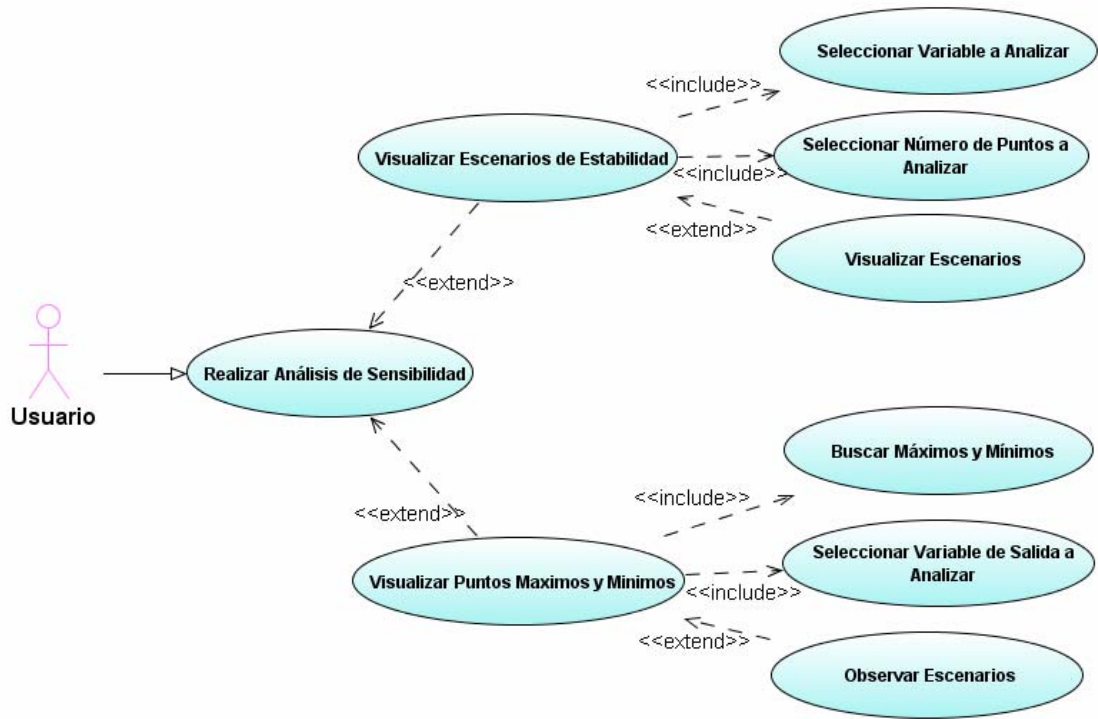


Figura 35. Diagrama Caso de Uso Realizar Análisis de Sensibilidad.

Id del Caso de Uso	006
Resumen:	Revisar estabilidad del modelo.
Actores Directos:	Usuario
Entradas:	<ol style="list-style-type: none"> 1. Variables Lingüísticas de entrada y salida. 2. Reglas de Inferencia.
Salidas:	<ol style="list-style-type: none"> 1. Escenarios de estabilidad. 2. Escenarios de Máximos y Mínimos.

Id del Caso de Uso	006
Precondiciones:	Modelo difuso completo (con variables de entrada, salida y reglas).
Escenario Principal de Éxito:	El modelo difuso está completo con todas sus variables y reglas.
Postcondiciones	El usuario obtiene conclusiones respecto al modelo.
Escenario de Extensiones Alternativas:	<ol style="list-style-type: none"> 1. Esquema general del modelo. 2. Variables Lingüísticas. 3. Reglas de capa 1 y 2. 4. Resultados capa 1 y 2.
Requerimientos especiales	<ol style="list-style-type: none"> 1. Definir variable de entrada para contrastar con las variables de salida. 2. Fijar puntos para visualizar escenarios de estabilidad. 3. Escoger variable de salida para adquirir datos de máximos y mínimos.
Criterios de aceptación	Permite al usuario observar los escenarios y datos de estabilidad y máximos y mínimos.

Tabla 10. Descripción Caso de Uso realizar Análisis de sensibilidad.

DIAGRAMAS DE ACTIVIDADES

Diagrama de Actividades Gestionar Variable Lingüísticas

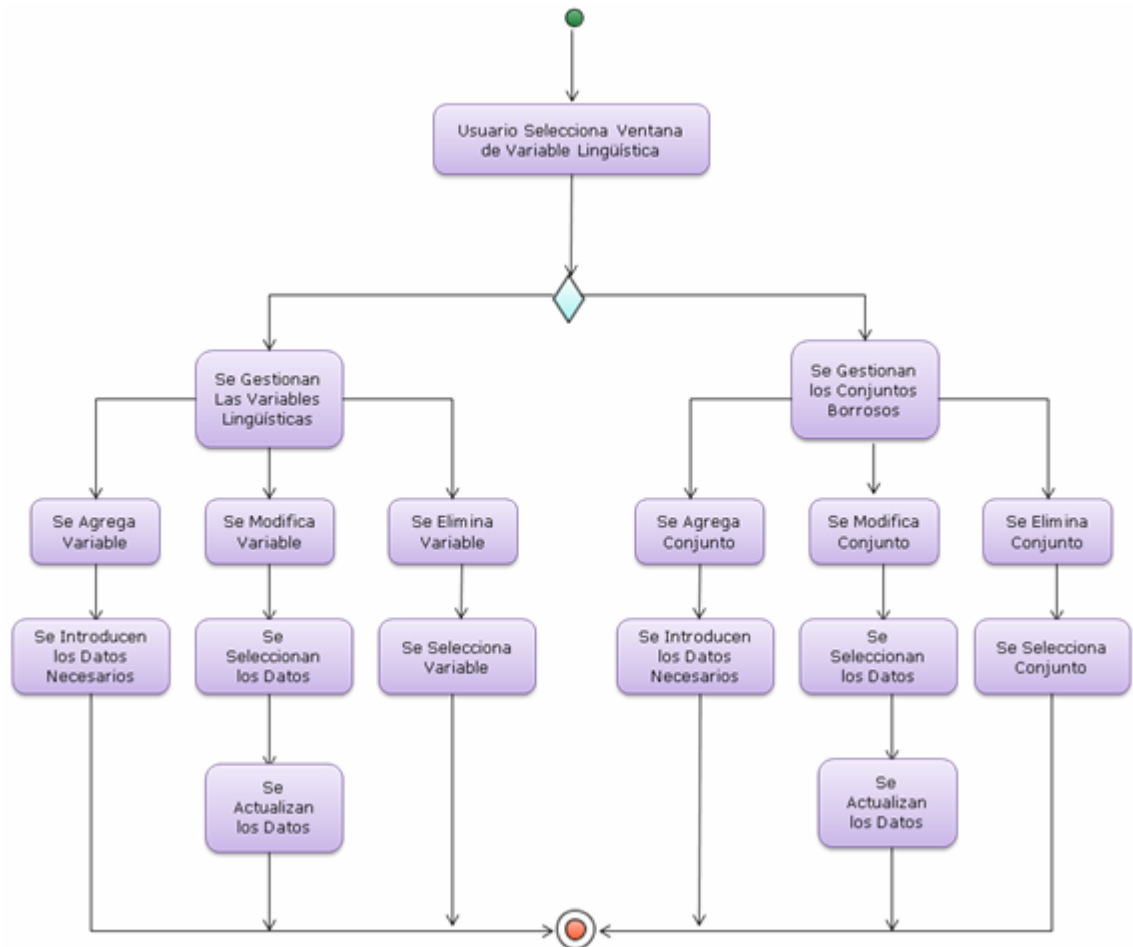


Figura 36. Diagrama de Actividades Gestionar Variables Lingüísticas.

Diagrama Actividades Gestionar Reglas

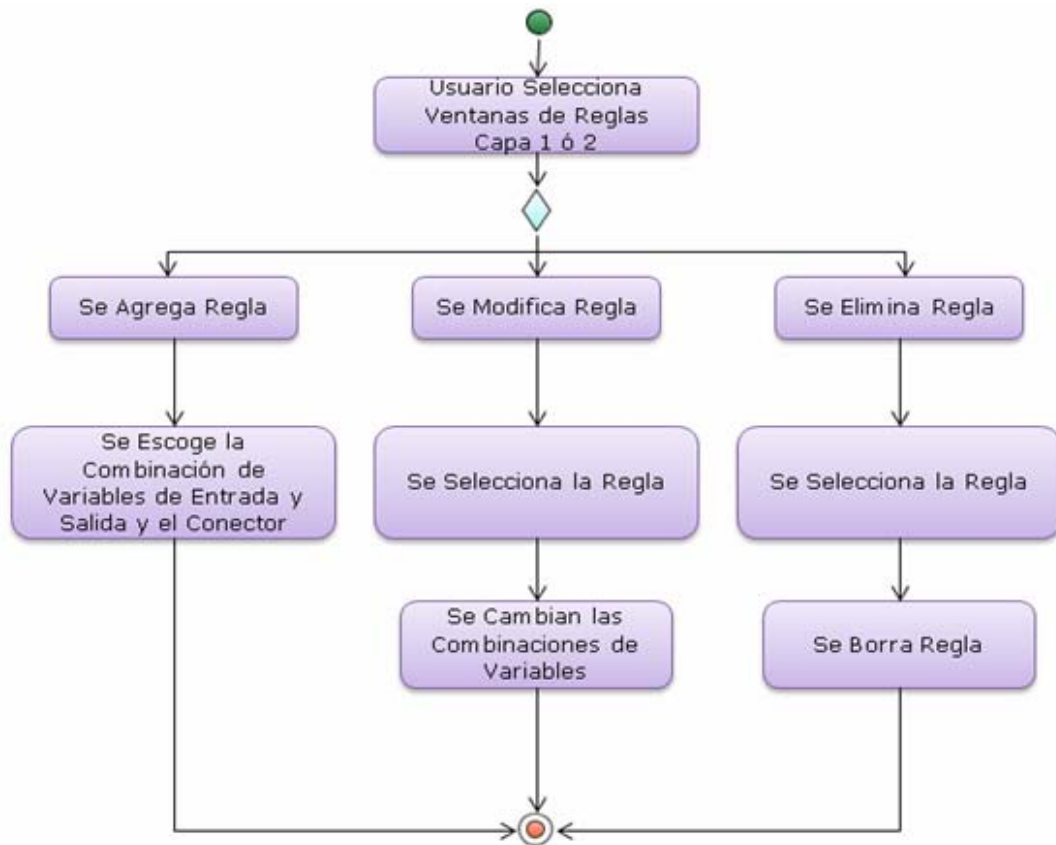


Figura 37. Diagrama de Actividades Gestionar Reglas.

Diagrama Actividades Visualizar Resultados

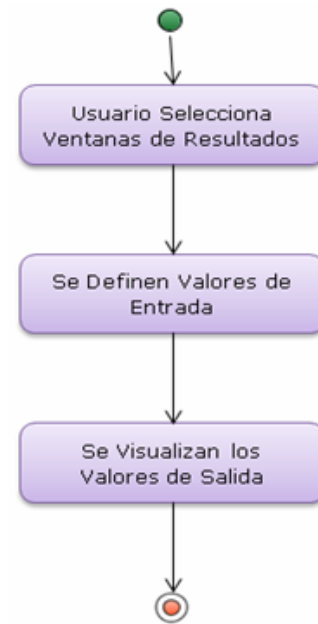


Figura 38. Diagrama de Actividades Visualizar Resultados.

Diagrama de Actividades Análisis de Sensibilidad

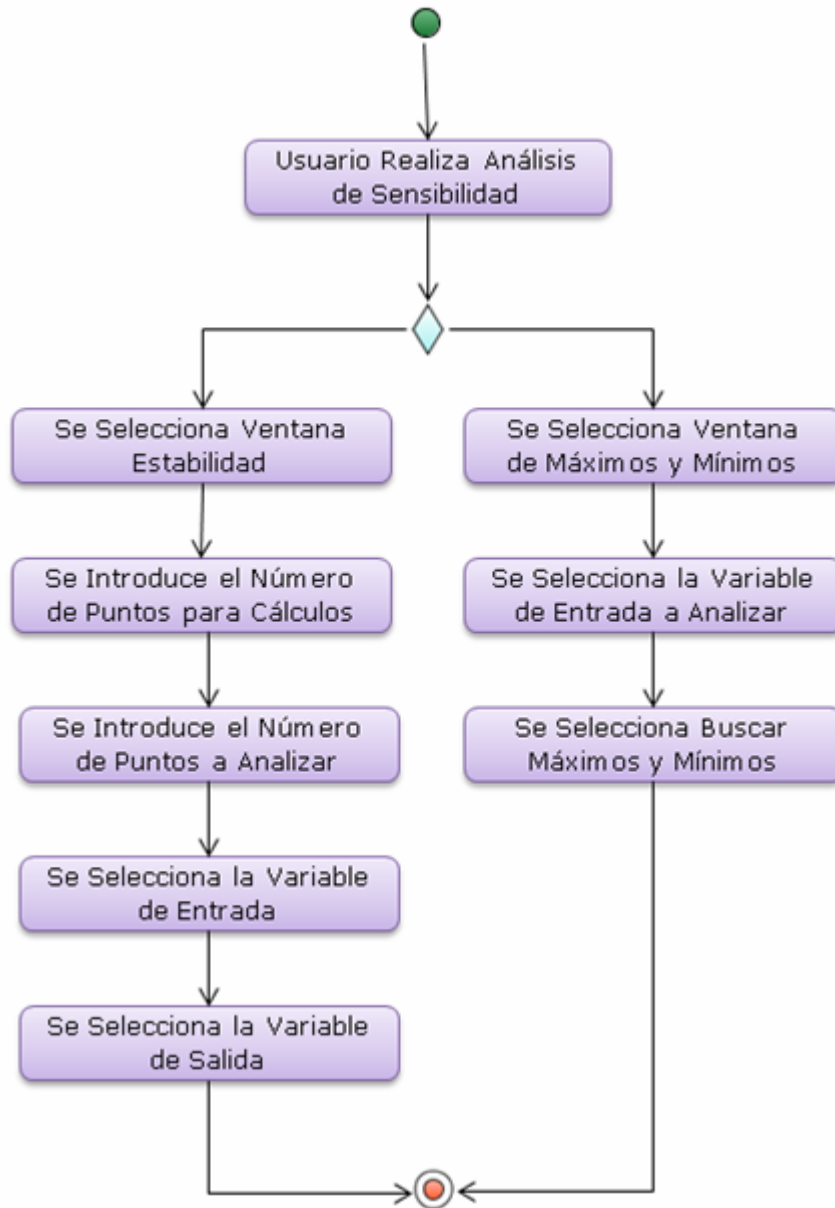


Figura 39. Diagrama de Actividades Análisis de Sensibilidad.

DIAGRAMAS DE CLASES

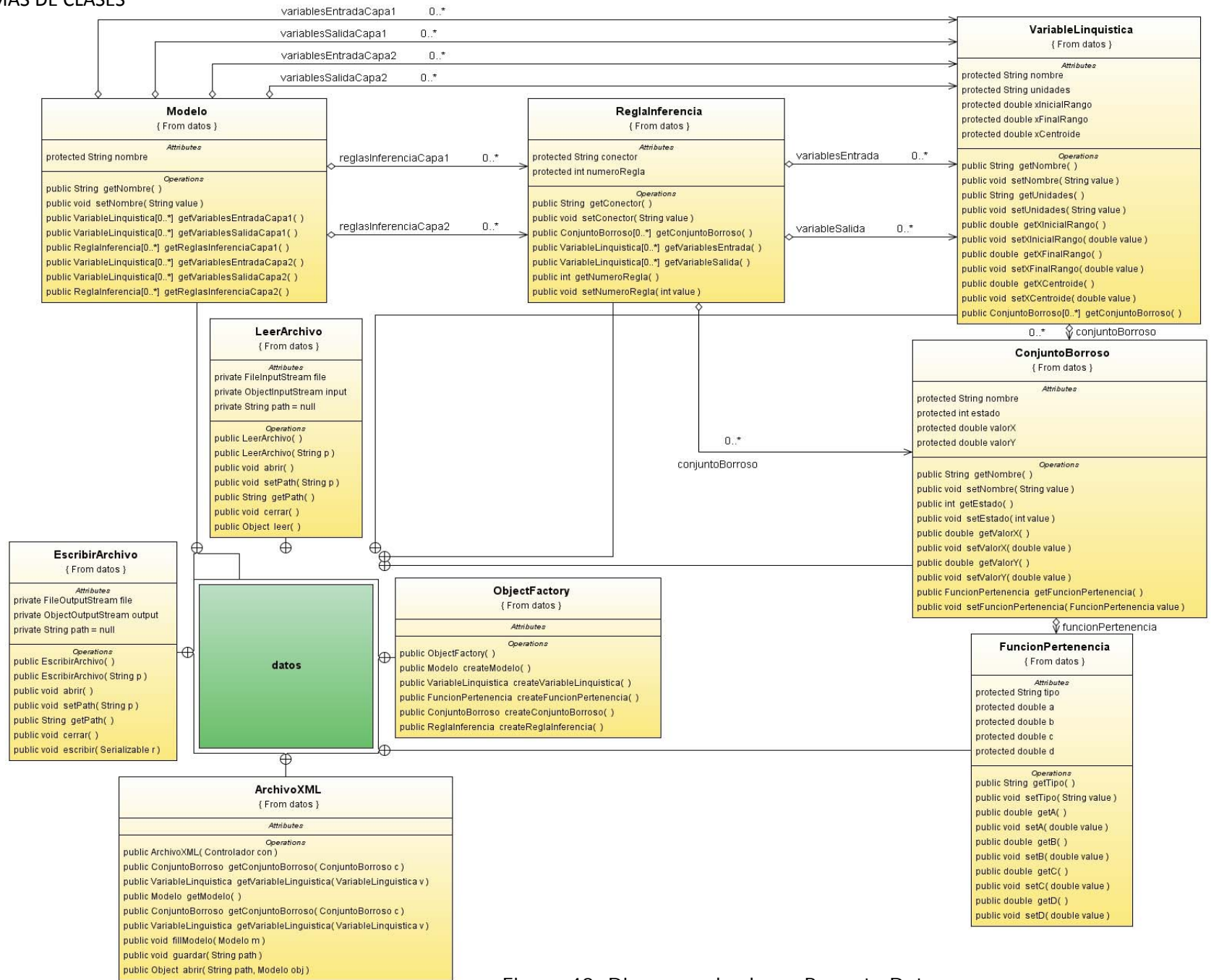


Figura 40. Diagrama de clases Paquete Datos.

3.2.3.2. DISEÑO DE LA INTERFAZ GRÁFICA

Se diseñó un entorno gráfico que fuera sencillo y de fácil entendimiento para el usuario para lo cual se pensó en dividir la aplicación en secciones que correspondan a cada uno de los pasos a través de los que se crea un modelo difuso, es decir, introducir sus variables lingüísticas con sus respectivos conjuntos difusos y crear las reglas de inferencia difusa, además de otras secciones en las que, se observan los resultados arrojados por un modelo luego de introducir un cierto valor para cada variable de entrada existente; se permite adelantar un análisis de sensibilidad basado en dos componentes, el primero donde se exhibe que tan estable es un modelo según algunos escenarios mostrados donde se puede ver la variación de la variable de salida con respecto a cualquiera de las variables de entrada, y segundo donde se puede reconocer los valores máximos y mínimos de las variables de salida; todo esto para que el usuario tome decisiones y concluya en base a los resultados obtenidos.

Según lo anterior a continuación se hace una descripción de cada una de las secciones que se incluirán en la aplicación y la forma como irán estructuradas, éstas secciones se organizan mediante pestañas en la ventana principal y al hacer click sobre alguna de ellas se da paso a la ventana correspondiente, adicionalmente se accede a las diferentes ventanas con un menú “modelo” que incluye todas las acciones pertinentes al mismo.

VENTANA PRINCIPAL ESQUEMA

Es la ventana donde se visualiza el esquema general de un modelo luego que éste es creado, allí se aprecian sus variables de entrada y salida, además del nombre con que ha sido guardado el modelo.

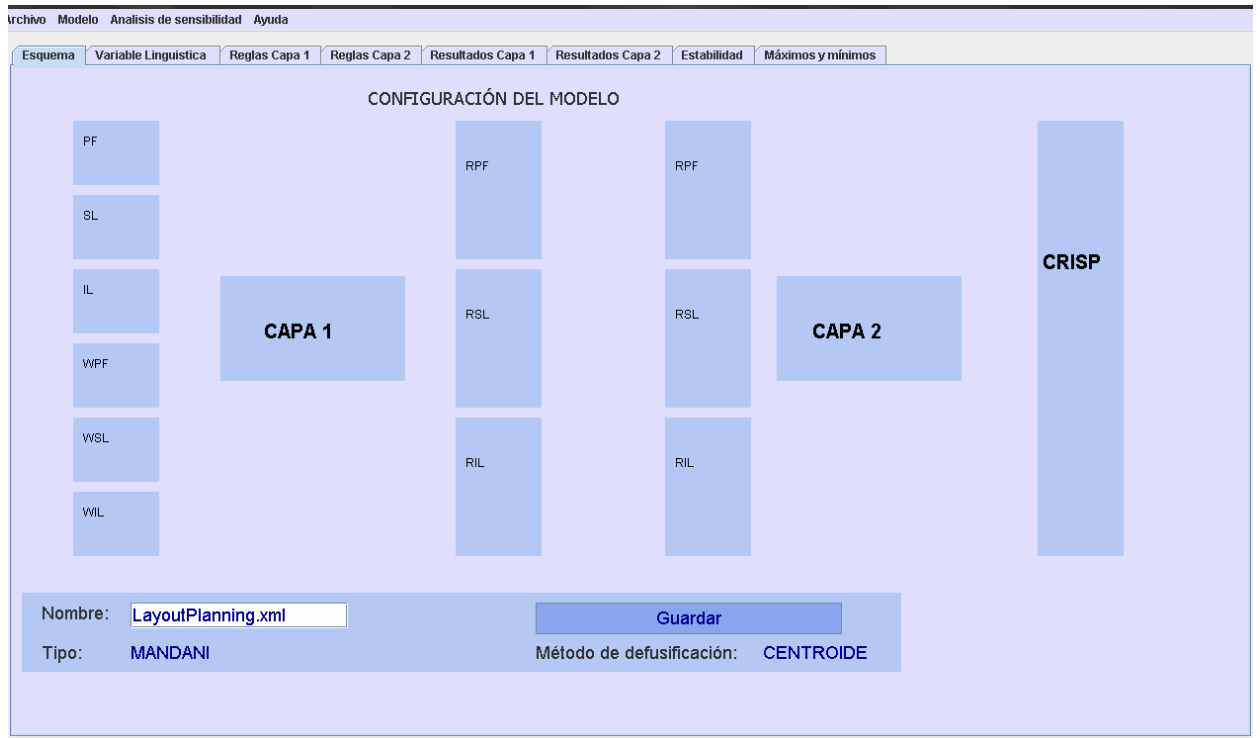


Figura 43. Ventana esquema.

VENTANA VARIABLE LINGÜÍSTICA

Se crean, modifican y eliminan variables lingüísticas, así como los conjuntos borrosos correspondientes a ellas, ésta ventana se encuentra dividida en dos bloques, uno para manejar lo referente a la variable, su nombre, unidades, dominio y lugar al que pertenece (variable de entrada o salida en la primera o segunda capa), el otro bloque para manejar lo relacionado a los conjuntos borrosos de una variable, nombre, función de pertenencia y parámetros de la función.

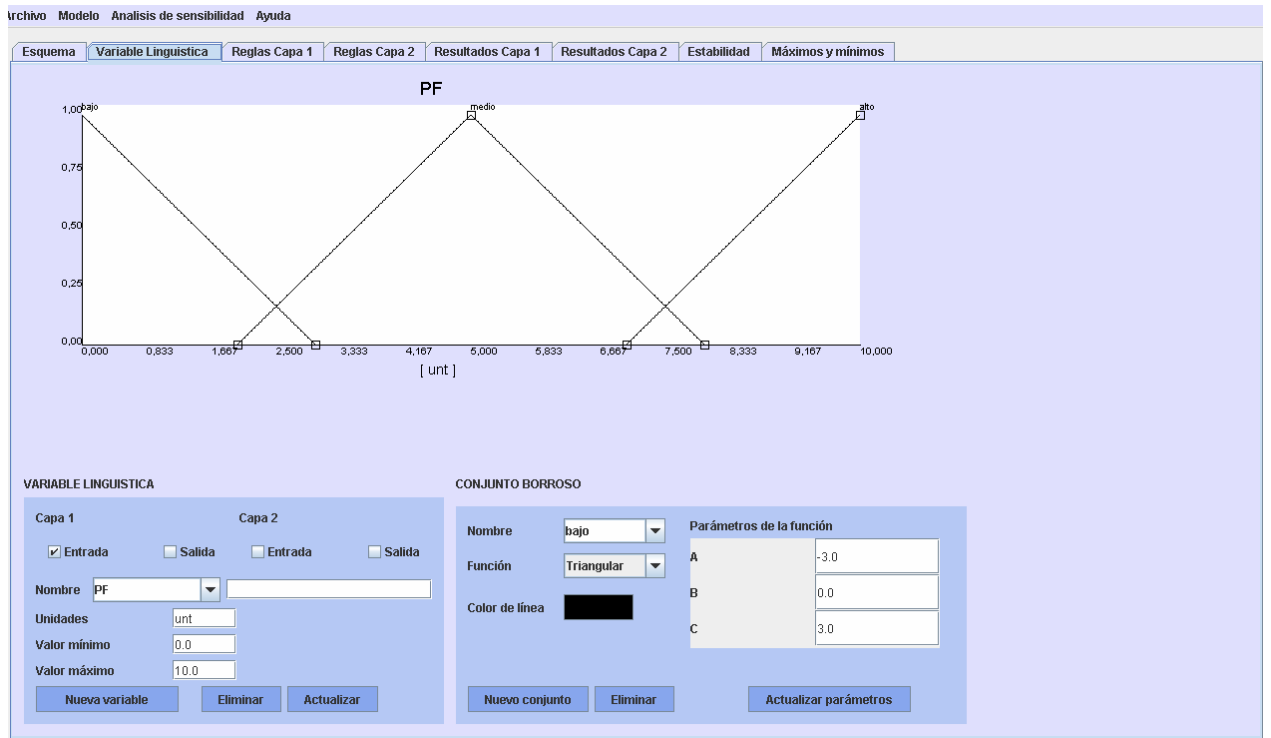


Figura 44. Ventana Variable Lingüística.

VENTANA REGLAS DE INFERENCIA

Para ésta sección se decidió crear dos pestañas independientes, una para las reglas referentes a la capa 1 del modelo y la otra para la capa 2; en ambas se crean, modifican y eliminan reglas de inferencia difusa donde se escogen las diferentes relaciones entre variables de entrada y salida, y el conector que se usa para esas relaciones que puede ser “Y” ó “O”.

Archivo Modelo Analisis de sensibilidad Ayuda

Esquema Variable Lingüística Reglas Capa 1 Reglas Capa 2 Resultados Capa 1 Resultados Capa 2 Estabilidad Máximos y mínimos

Reglas de Inferencia

1. SI (PF ES bajo) AND (WPF ES BAJO) ENTONCES (RPF ES MUY BAJO)
 2. SI (PF ES bajo) AND (WPF ES MEDIO) ENTONCES (RPF ES BAJO)
 3. SI (PF ES medio) AND (WPF ES MEDIO) ENTONCES (RPF ES MEDIO)
 4. SI (PF ES alto) AND (WPF ES MEDIO) ENTONCES (RPF ES ALTO)
 5. SI (PF ES alto) AND (WPF ES MEDIO) ENTONCES (RPF ES MUY ALTO)
 6. SI AND (SL ES bajo) AND (WSL ES BAJO) ENTONCES (RSL ES MUY BAJO)
 7. SI AND (SL ES bajo) AND (WSL ES MEDIO) ENTONCES (RSL ES BAJO)

Entradas

PF SL IL WPF WSL WIL

bajo	bajo	bajo	BAJO	BAJO	BAJO
medio	medio	medio	MEDIO	MEDIO	MEDIO
alto	alto	alto	ALTO	ALTO	ALTO
NINGUNO	NINGUNO	NINGUNO	NINGUNO	NINGUNO	NINGUNO

Entonces

RPF RSL RIL

MUY BAJO ▲	MUY BAJO ▲	MUY BAJO ▲
BAJO ▢	BAJO ▢	BAJO ▢
MEDIO ▣	MEDIO ▣	MEDIO ▣
ALTO ▼	ALTO ▼	ALTO ▼

Conector

Y O

Borrar Agregar

Figura 45. Ventana Reglas Capa 1.



Figura 46. Ventana Reglas Capa 2.

VENTANA RESULTADOS

Ésta sección se divide en dos pestañas que dan paso a dos ventanas independientes, la primera para los resultados de la capa 1 y la segunda para los de la capa 2, allí se le puede dar a las variables un valor de entrada de forma gráfica o manual, cabe resaltar que esto solo es posible para la capa 1 ya que sus salidas constituyen variables de entrada para la capa 2; siendo en esta ultima donde se aprecia el resultado final del proceso y la visualización de las variables de salida.

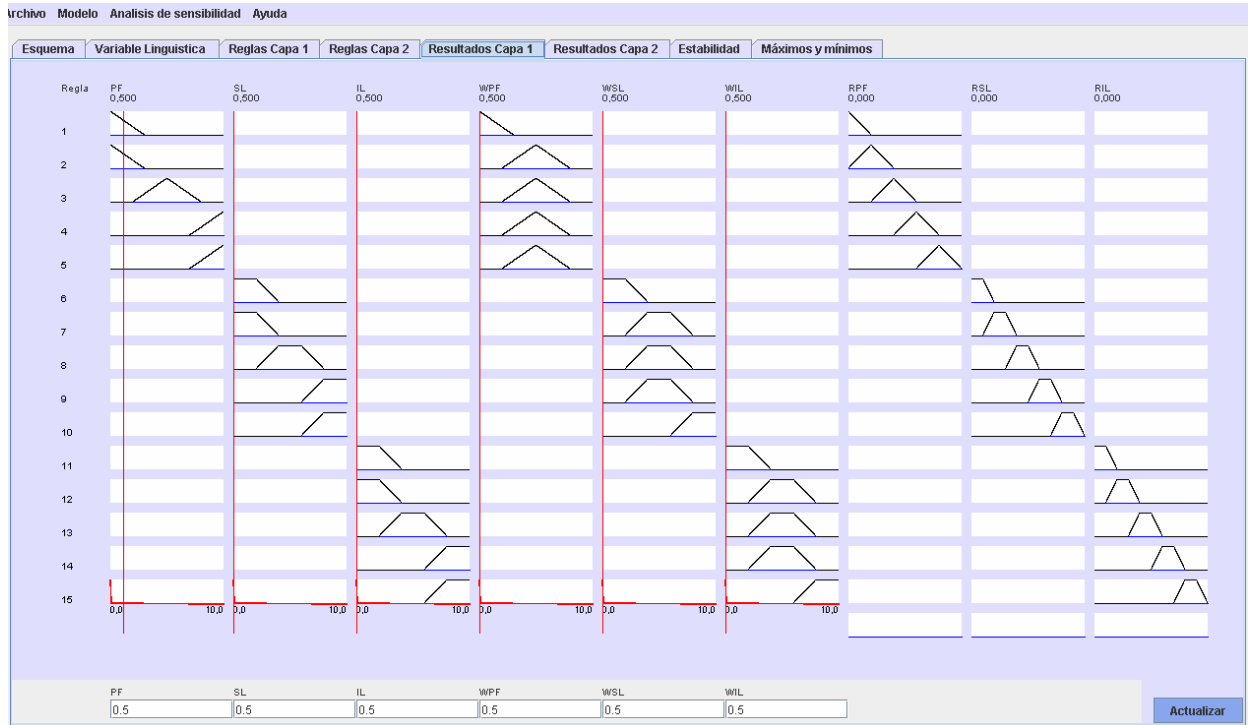


Figura 47. Ventana Resultados Capa 1.

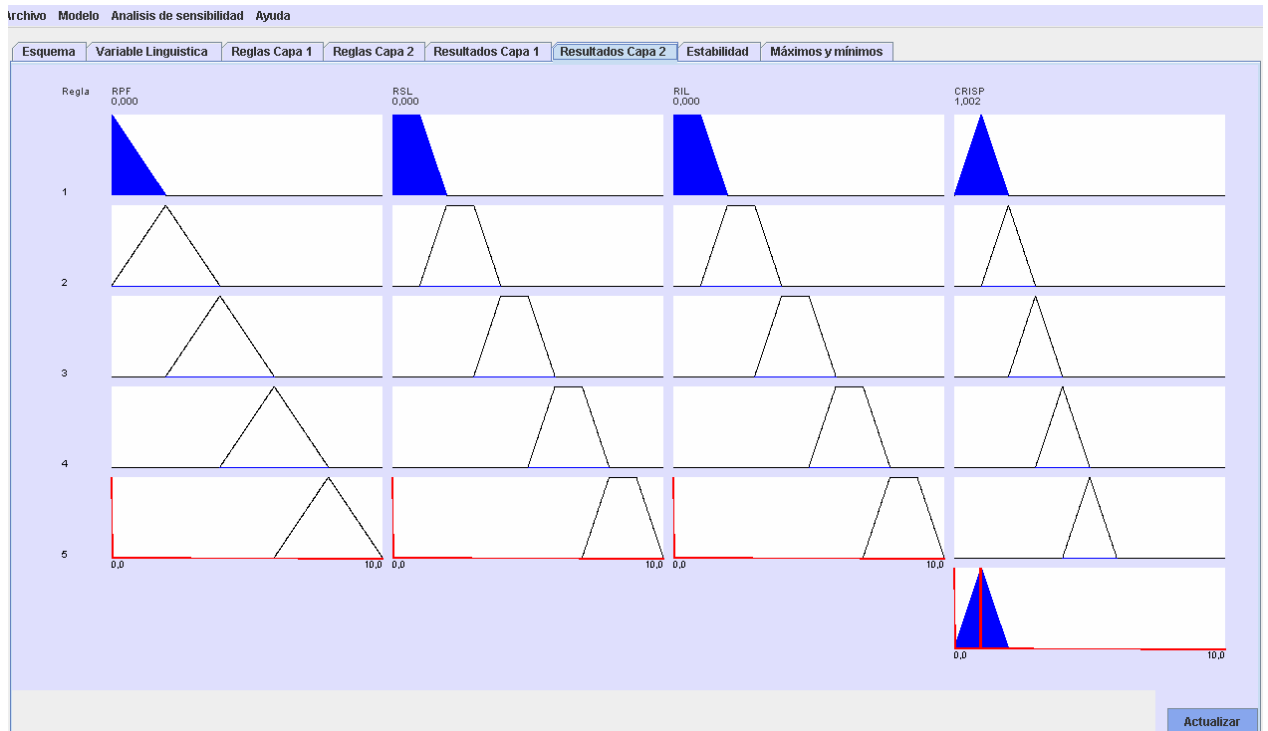


Figura 48. Ventana Resultados Capa 2.

VENTANA ANÁLISIS DE SENSIBILIDAD

Ésta sección se divide en dos pestañas, la primera, para visualizar la estabilidad de un modelo, esto se hace a través de escenarios gráficos y con datos, seleccionando la cantidad de puntos según los cuales se quiere analizar el modelo y la variable de entrada a través de la que se desea observar el comportamiento de la salida, la segunda, para ver los máximos y mínimos de la salida de un modelo, esto se hace a través de escenarios o cuadros que muestren los datos de los valores de las entradas y el valor máximo o mínimo encontrado según éstos, se exhiben 10 escenarios para los máximos y 10 para los mínimos.

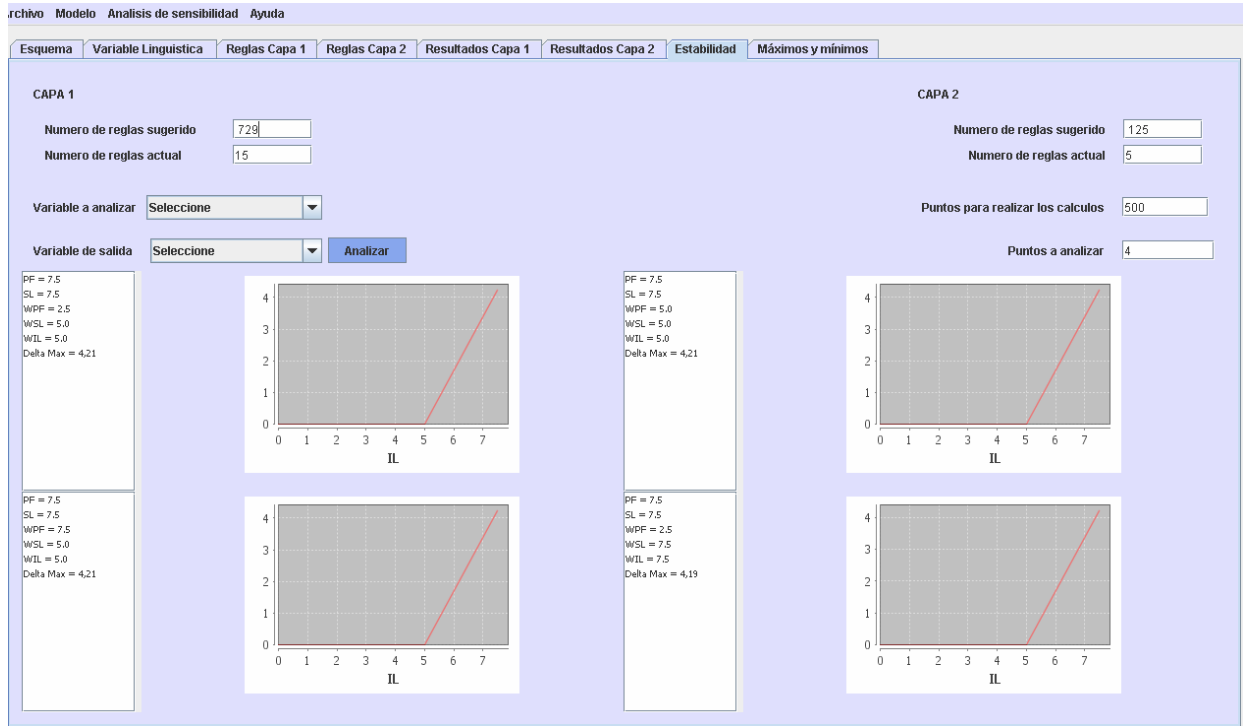


Figura 49. Ventana Estabilidad

Archivo Modelo Analisis de sensibilidad Ayuda

Esquema Variable Lingüística Reglas Capa 1 Reglas Capa 2 Resultados Capa 1 Resultados Capa 2 Estabilidad Máximos y mínimos

Buscar máximos y mínimos Variable a analizar CRISP

MÁXIMOS

PF = 7,50 SL = 7,50 IL = 7,50 WPF = 2,50 WVSL = 5,00 WML = 5,00 CRISP = 4,22	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 5,00 WVSL = 5,00 WML = 5,00 CRISP = 4,22	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 7,50 WVSL = 5,00 WML = 5,00 CRISP = 4,22	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 2,50 WVSL = 7,50 WML = 7,50 CRISP = 4,20	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 5,00 WVSL = 7,50 WML = 7,50 CRISP = 4,20
PF = 7,50 SL = 7,50 IL = 7,50 WPF = 7,50 WVSL = 7,50 WML = 7,50 CRISP = 4,20	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 2,50 WVSL = 7,50 WML = 5,00 CRISP = 4,14	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 2,50 WVSL = 5,00 WML = 7,50 CRISP = 4,14	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 5,00 WVSL = 7,50 WML = 5,00 CRISP = 4,14	PF = 7,50 SL = 7,50 IL = 7,50 WPF = 5,00 WVSL = 5,00 WML = 7,50 CRISP = 4,14

MÍNIMOS

PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 7,50 WML = 2,50 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 7,50 WML = 5,00 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 7,50 WML = 5,00 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 2,50 WML = 2,50 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 2,50 WML = 5,00 CRISP = 0,00
PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 2,50 WML = 7,50 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 5,00 WML = 2,50 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 5,00 WML = 2,50 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 7,50 WVSL = 5,00 WML = 7,50 CRISP = 0,00	PF = 7,50 SL = 7,50 IL = 2,50 WPF = 2,50 WVSL = 7,50 WML = 2,50 CRISP = 0,00

Figura 50. Ventana Máximos y Mínimos.

3.3. FASE DE CONTRUCCIÓN

En ésta fase se desarrolla específicamente la codificación de las acciones que debe realizar la aplicación, entre las cuales podemos destacar la implementación de lógica para la creación de los modelos, la parametrización, los algoritmos para el manejo de variables lingüísticas, evaluación de reglas, defusificación y análisis de sensibilidad; del éxito de ésta fase depende la ejecución y desarrollo de la herramienta y se asegura que se cumpla con los requisitos definidos.

3.3.1. PLANEACIÓN DE LA FASE:

Se determina tener en cuenta aspectos que rigen la puesta en marcha de la codificación de cada una de las componentes que posee el software, utilizando

por ejemplo métodos teóricos como el del centroide para encontrar la salida final del modelo, o la utilización de funciones de pertenencia regidas por parámetros que son parte fundamental en la creación de variables lingüísticas.

A continuación se presentan las actividades propuestas para esta etapa:

Realizar el proceso de implementación de la herramienta, teniendo en cuenta cada uno de los elementos que permiten definir su estructura.

Verificar el cumplimiento de los requisitos propuestos.

Construcción del prototipo.

3.3.2. SOFTWARE Y LENGUAJE DE PROGRAMACIÓN UTILIZADO

Para elaborar el prototipo de herramienta se determinó la utilización de los siguientes recursos:

Lenguaje Java

Sistema Operativo Windows XP

Netbeans

3.3.3. DESCRIPCIÓN DE LOS COMPONENTES DE LA APLICACIÓN

Se presenta una descripción generalizada de los elementos que integran la aplicación, con el fin de dar a conocer los aspectos más relevantes en la elaboración de la herramienta.

3.3.3.1. IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA

La implementación de la interfaz gráfica se realizó separadamente de la lógica y de datos que componen la aplicación, esto encapsulándola dentro de un paquete llamado presentación; dentro de este paquete se encuentran varias clases como PanelFuncionPertenencia, un objeto de ésta clase se encarga de la funcionalidad del panel de parámetros de las funciones de pertenencia, así como de la representación de los conjuntos difusos en el sector del gráfico de las variables lingüísticas al manipular las unidades de dispositivo correspondientes a los pixeles de la pantalla y las unidades lógicas correspondientes a las unidades de cada variable, FrmPrincipal, un objeto de ésta clase gestiona el inicio de la aplicación, así mismo dentro de éste paquete de presentación se manejan los paneles de variable, reglas, resultados, estabilidad y máximos y mínimos.

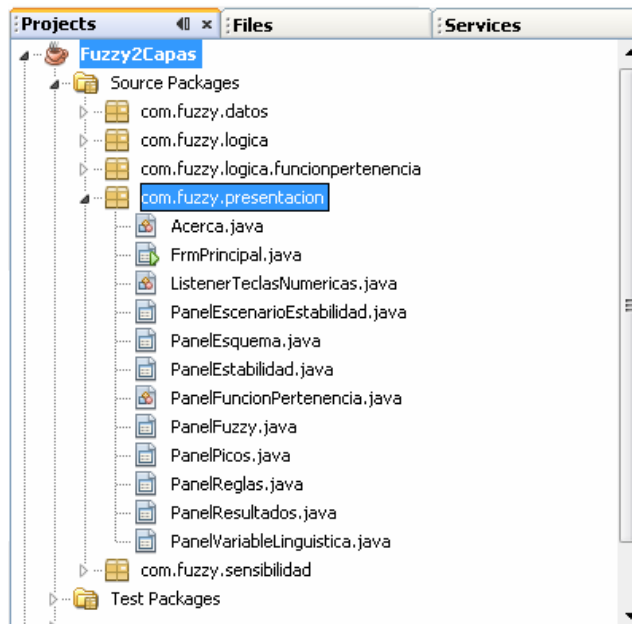


Figura 51. Árbol de Directorio de presentación en el paquete de Información Presentación.

3.3.3.2. IMPLEMENTACIÓN DEL MODELO VISTA CONTROLADOR

Como se menciona en el capítulo 2, el patrón modelo vista controlador es un patrón diseñado para separar los datos de una aplicación, la interfaz gráfica y la lógica de control con el objetivo de reutilizar componentes fácilmente, cada una de estas partes se implementaron en el paquete `com.fuzzy.lógica`; el controlador se encuentra presente en todas las clases encargadas de la interfaz gráfica y se podría decir que su función es la de “monitorear” que las acciones realizadas por el usuario se reflejen en acciones del sistema, entre estas acciones se podrían resaltar el manejo de escenarios de estabilidad y puntos máximos y mínimos en el paquete `Sensibilidad`, la comunicación con la vista para inicializar la presentación de las ventanas requeridas por el usuario a través de la clase “`FrmPrincipal`”, la relación con el modelo²⁴ para todo lo referente a los datos, la gestión de las funciones de pertenencia incluidas en la herramienta y que se encuentran en el paquete `Funcionpertenencia` entre otras; la vista hace referencia a la presentación gráfica e incluye paneles como el de esquema, reglas y estabilidad por nombrar algunos y que se encuentran en el paquete `Presentación`, por último el modelo referencia la información con el que opera la herramienta, básicamente los datos que se encuentran en las variables de entrada y salida de cada una de las capas, reglas y parámetros establecidos entre otros, por esta razón su función está direccionada al paquete `Datos`.

²⁴ Modelo referente al Modelo Vista Controlador MVC

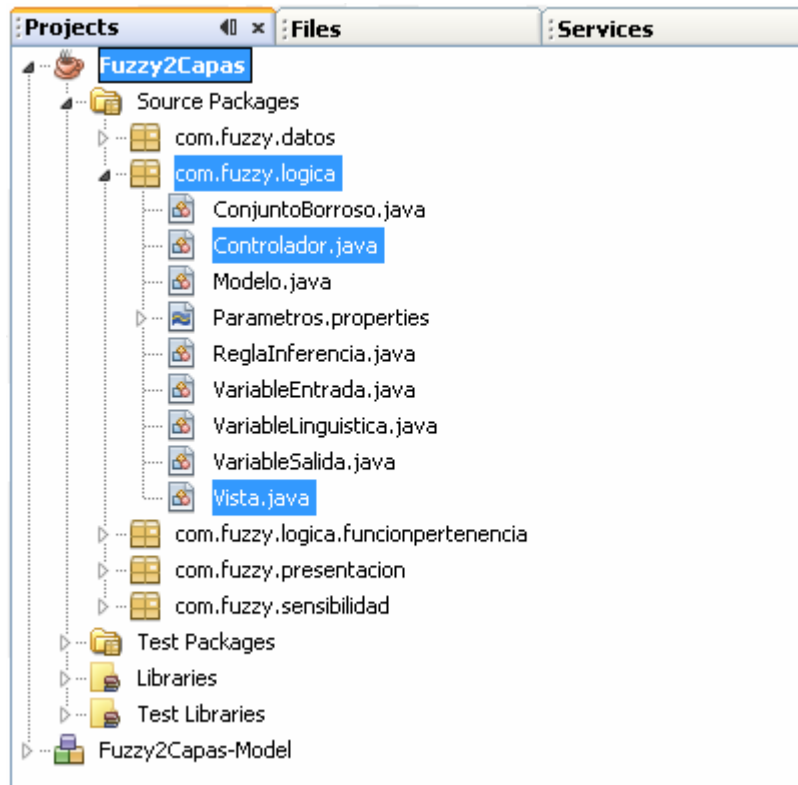


Figura 52. Árbol de Directorio del Modelo Vista Controlador MVC en el Paquete de Información Lógica.

3.3.3.3. MANIPULACIÓN DE VARIABLES LINGÜÍSTICAS

El manejo de las variables lingüísticas se representa en dos etapas: la primera en donde se ingresan los parámetros necesarios para la creación de variables lingüísticas, conjuntos borrosos y los valores de entrada y de salida, esto se encuentra integrado dentro del paquete Lógica que contiene las clases: “variable Lingüística” y que se encarga de acciones como el de dar nombre, establecer el dominio, introducir el conjunto borroso y las unidades de la variable; “Conjunto Borroso” en donde se asigna el nombre al conjunto borroso, se establecen las funciones de pertenencia asociadas, y se determinan las coordenadas para

graficar el conjunto; “variable Entrada” y “variable Salida” en donde quedan establecidas las características de cada una de ellas, y la segunda en donde se representan los valores ingresados, este se encuentra en el paquete Datos y en él se referencian las clases “variable lingüística” y “conjunto borroso” encargadas de dicho proceso.

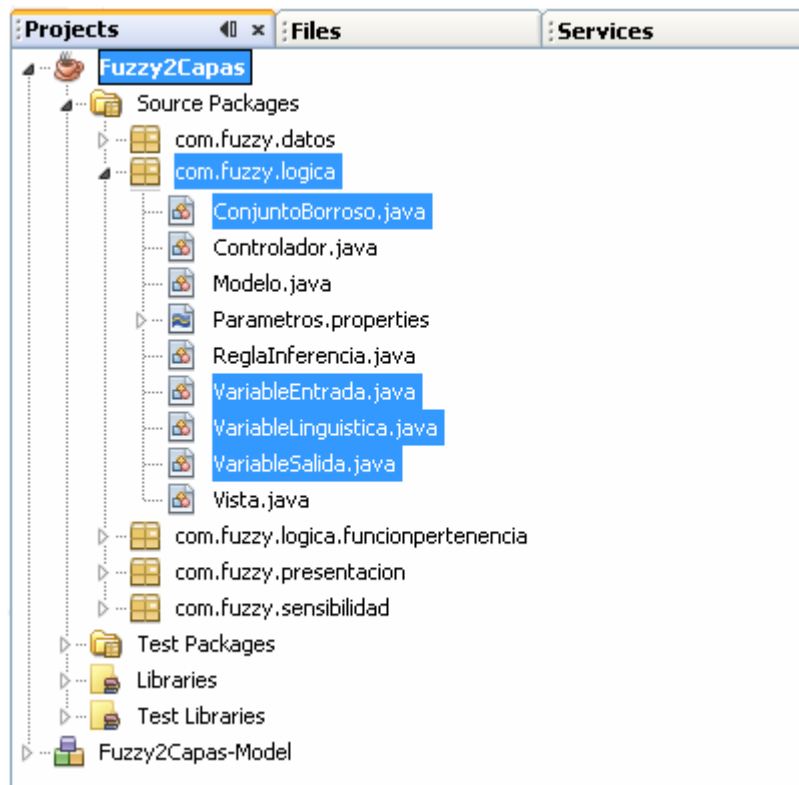


Figura 53. Árbol de directorio de la variable Lingüística, Conjunto Borrosos y Variables de Entrada y Salida en el Paquete de Información Lógica.

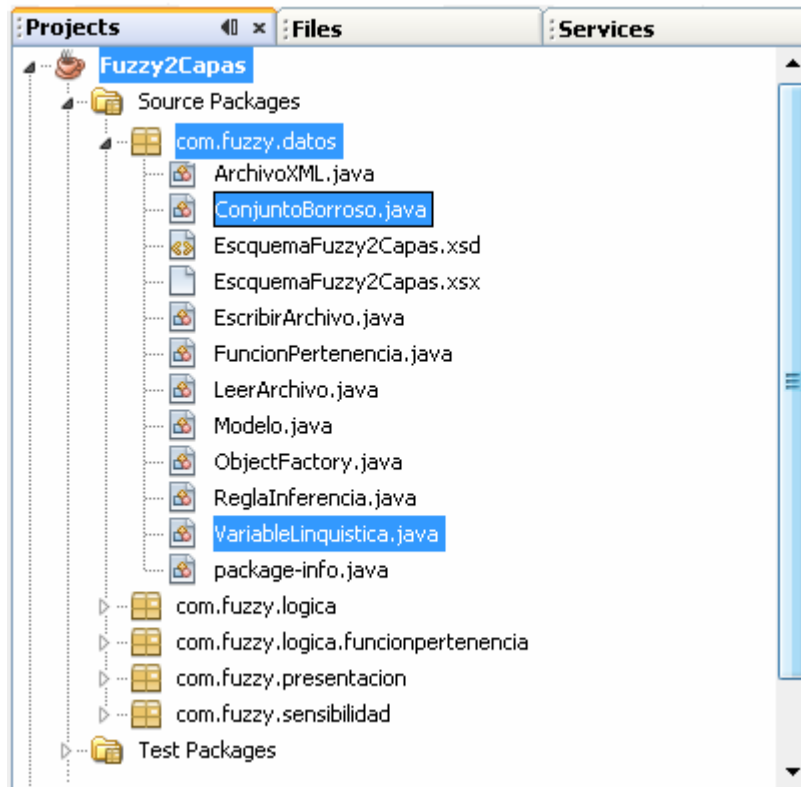


Figura 54. Árbol de directorio de la variable Lingüística y Conjunto Borroso en el Paquete de Información Datos.

3.3.3.4. EVALUACIÓN DE REGLAS

La evaluación de reglas constituye la forma como los valores lingüísticos son generados; para la construcción de esta etapa, se generó una clase denominada “regla inferencia” que se encuentra dentro del paquete Lógica y que se encarga del manejo de las reglas de inferencia; entre sus funciones se encuentran establecer los parámetros de los conjuntos borrosos y las variables de entrada y salida, definir los conectores “Y” u “O” y el orden asociado a la creación de las reglas, así como la eliminación de un conjunto de inferencia entre otras cosas.

La representación de los datos requeridos se realiza a través de la clase “regla de inferencia” que se cita en el paquete Datos, y que retorna dichos valores.

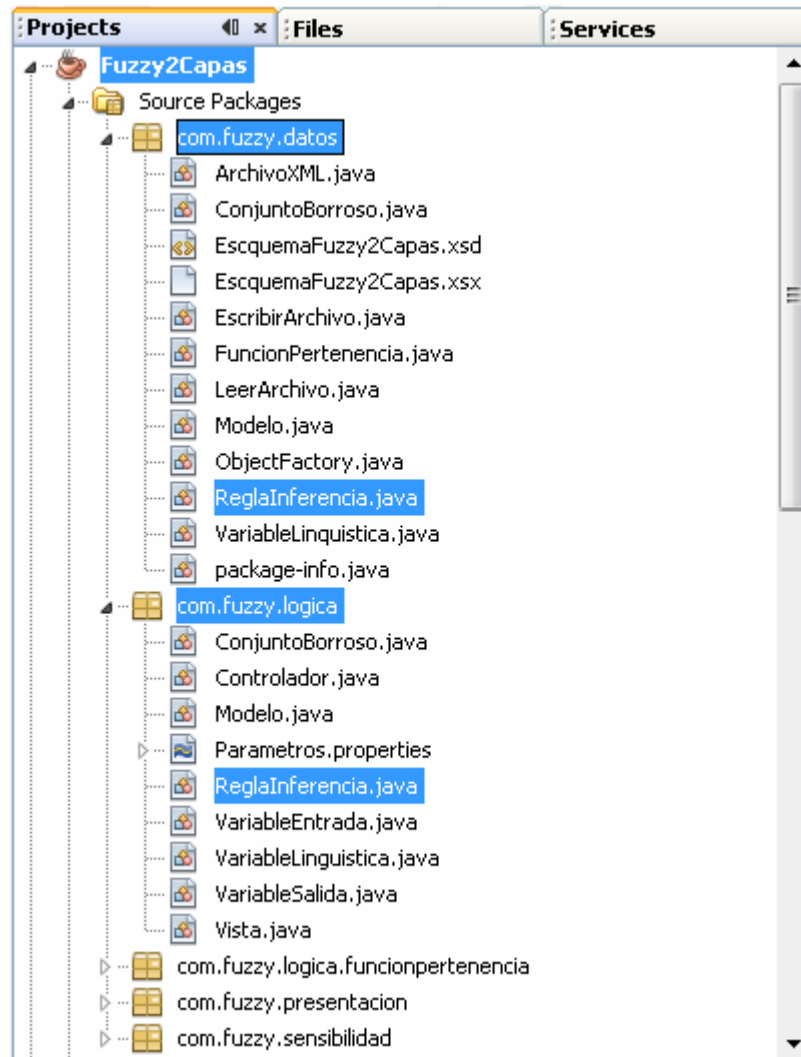


Figura 55. Árbol de directorio de la regla de inferencia en los Paquetes de Información Datos y Lógica.

3.3.3.5. IMPLEMENTACIÓN DE LAS FUNCIONES CARACTERÍSTICAS O DE PERTENENCIA

Las funciones características representan la pertenencia de un elemento a un conjunto, para la construcción de esta etapa se limitó la utilización de 5 funciones a saber: Gaussiana, Curva de Bell, Sigmoïdal, Trapezoidal y Triangular ya que son las más utilizadas, estas funciones fueron implementadas en el paquete denominado Funcionpertenencia el cual contiene una interface denominada "FuncionPertenencia" y que tiene como función establecer los criterios para definir el comportamiento y la representación grafica de las mismas; y cinco clases que representan cada uno de los tipos de función descrita anteriormente, en cada una de estas clases se establece el dominio y los parámetros requeridos de acuerdo a las características que poseen; en el paquete datos se presenta una interfaz "FuncionPertenencia" que contiene los datos que hacen referencia a las funciones y que son invocados a través de objetos pertenecientes a dicha clase.

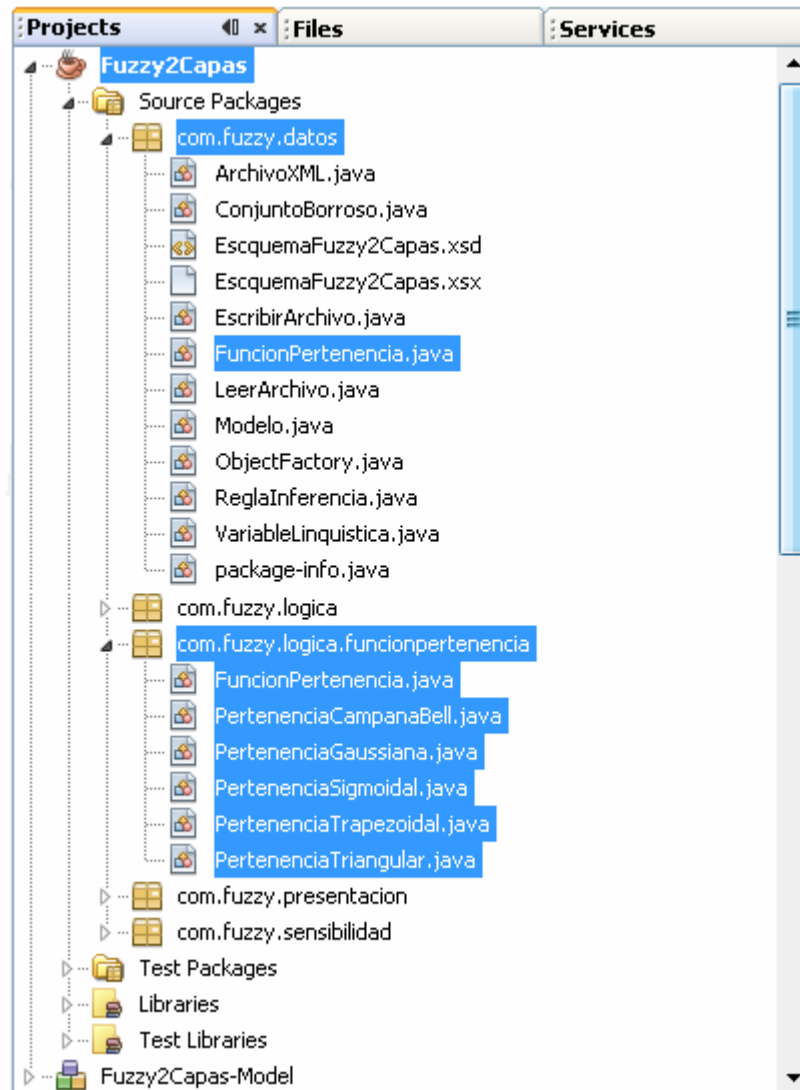


Figura 56. Árbol de Directorio de Funciones de Pertenencia en los Paquetes de información Datos y Lógica.

3.3.3.6. ALGORÍTMOS DE DEFUSIFICACIÓN

El algoritmo de defusificación fue implementado tomando los resultados difusos obtenidos del bloque de inferencia y calculando el resultado numérico para cada

variable de salida por medio del método del centro de área o centroide. Lo primero para hallar éste valor es encontrar el área total aproximada de la figura difusa, resultante de la inferencia, dividiéndola en intervalos y haciendo uso del método numérico Simpson 1/3 en cada uno de ellos, una cantidad de 1000 intervalos se utiliza para realizar éste procedimiento, luego se procede a encontrar el centroide de la figura; éste procedimiento conlleva un tiempo de ejecución considerable a la aplicación.

3.3.3.7. ALGORÍTMOS DE ANÁLISIS DE SENSIBILIDAD

El análisis de sensibilidad se realiza basado en dos componentes, uno es una muestra de la estabilidad del modelo, exponiendo los escenarios en los que se presentan mayores cambios a sus variables de salida en contraste con la modificación de los valores de alguna de sus variables de entrada; el otro consiste en mostrar los 10 escenarios de las salidas en los que se presentan los picos más altos y los 10 en los que están los más bajos, cabe resaltar que éste análisis solo se hace para modelos difusos de dos capas y no de una sola, ya que las salidas de la capa 1 dependen directamente de las entradas, por el contrario, en uno de dos capas las salidas dependen de cómo se relacionan las variables de entrada de la primera y segunda capa.

El primer componente es un algoritmo de estabilidad, como su nombre lo indica comprueba que tan estable es un modelo según cambios en su salida respecto a modificaciones en una de sus entradas, se desarrolló definiendo una variable de entrada para contrastarla con una de las variables de salida especificando un número de puntos según los cuales se discretizan todas las variables y se analizan los resultados del modelo. La variable de entrada, salida y el número de puntos para la discretización son escogidos por el usuario. El algoritmo crea escenarios iterando sobre los puntos definidos para dar un valor fijo a las variables

de entrada que no fueron escogidas y comienza a iterar sobre esos mismos puntos en la variable a analizar, es decir que en un escenario el valor de esta variable va cambiando mientras los de las demás permanecen fijos, de esa forma se puede apreciar que alteraciones se producen en la salida del modelo en base a los cambios producidos en una de sus entradas; para dar lugar al siguiente escenario el algoritmo itera de nuevo y hace un cambio en las demás variables de entrada modificando sus valores, estos permanecen fijos mientras se itera sobre la variable escogida, y así sucesivamente se va dando origen a todos los escenarios que se van almacenando dentro de un vector. Cada escenario está compuesto por los valores de las variables de entrada que permanecen fijos, el máximo delta en la salida y una gráfica que muestra la variación de la salida del modelo contra la entrada escogida. Para encontrar el delta máximo de cada escenario, cada vez que el algoritmo itera sobre la variable de entrada va comparando el valor de la salida con el obtenido de la iteración anterior y examina cuál de los dos es el mayor, de esa forma al finalizar las iteraciones en ese escenario el valor del máximo de los deltas estará guardado; el objetivo de hallar el máximo delta en un escenario es encontrar cambios de altura en la gráfica de la entrada vs la salida para determinar que tan estable es el comportamiento del modelo frente a modificaciones en uno de los parámetros que lo afectan, éste algoritmo requiere un tiempo prudencial para su realización, por lo que puede retardar la ejecución de la aplicación dependiendo de la cantidad de variables incluidas en el modelo y de la cantidad de reglas existentes.

El segundo componente, algoritmo de máximos y mínimos, se encarga de hallar y mostrar los 10 mayores máximos y los 10 menores mínimos; su funcionamiento es similar al del algoritmo de estabilidad, inmediatamente anterior. Este crea escenarios compuestos por los valores de las variables de entrada y un máximo valor de la variable de salida o un mínimo según sea el caso. Como un modelo de dos capas puede tener más de una variable de salida, el usuario escoge para cuál

de ellas quiere ver los máximos y mínimos y el algoritmo los encuentra según la variable. El algoritmo se divide en dos partes una para los escenarios de máximos y otra para los de mínimos, los escenarios de los máximos son creados haciendo iteraciones sobre las variables de entrada, el algoritmo calcula el valor de salida para esa combinación de valores de entrada y lo compara con el hallado en la anterior iteración para determinar el mayor de los dos, así al final de las iteraciones los máximos quedan almacenados y se muestran los 10 escenarios correspondientes a los mayores máximos con los respectivos valores de entrada que dieron origen a ese punto máximo. El mismo procedimiento es realizado para encontrar y mostrar los 10 escenarios de los menores mínimos, teniendo en cuenta los valores mínimos de salida en lugar de los máximos.

3.3.3.8. IMPLEMENTACIÓN DE PERSISTENCIA DE DATOS

Para la persistencia de datos se recurrió a la serialización de los mismos y a la tecnología xml, la serialización de los objetos se realiza por medio de la implementación de la interfaz serializable en cada clase de este tipo, dentro del paquete Lógica, esto se hace con el propósito de informar a la JVM (Java Virtual Machine) que un objeto va a ser serializado, las clases utilizadas para conformar un modelo difuso fueron: “conjunto borroso”, “regla inferencia”, “variable lingüística”, “variable entrada”, “variable salida”, y las clases relacionadas con las funciones de pertenencia de un conjunto borroso; dentro del paquete Datos también se implementaron clases para el manejo de conjuntos borrosos, función de pertenencia, modelo, regla de inferencia y variable lingüística, generadas a partir del API JAXB por medio del recurso XMLSchema para construir los esquemas XML de cada una y de esa forma guardar y abrir archivos en la aplicación; adicionalmente se implementa una clase de Java llamada “Object Factory” dentro del mismo paquete para construir nuevas instancias de representación para contenido XML, de manera que se puedan crear nuevos

documentos XML basados en esquemas ya creados y así crear nuevos archivos dentro de la aplicación.

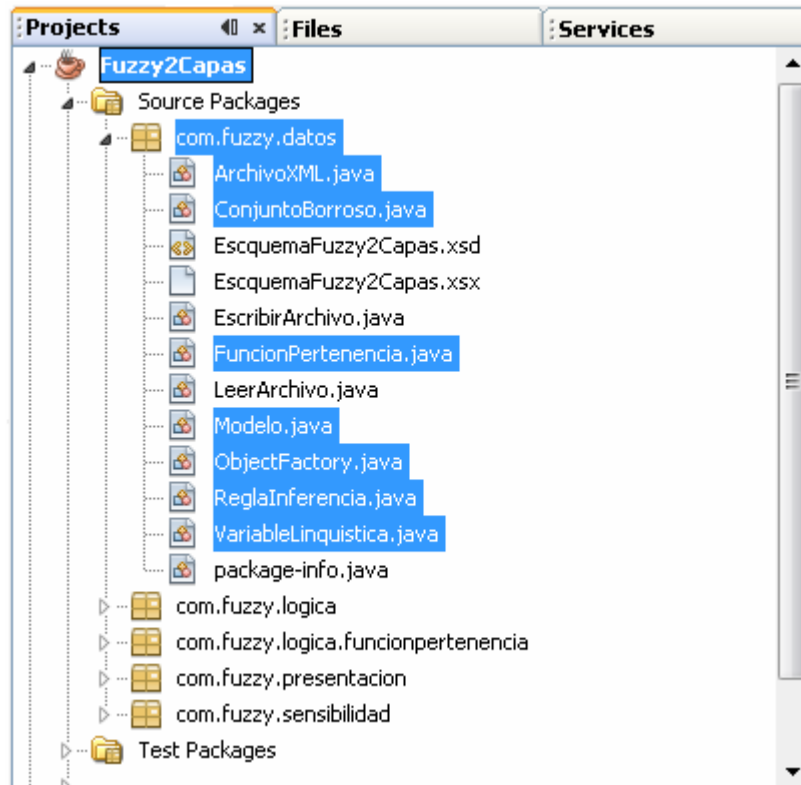


Figura 57. Árbol de directorio de la Manipulación de la Persistencia de Datos en el Paquete de Información Datos.

A continuación se presentan los diagramas XSD que representan la forma como se almacenan los datos:

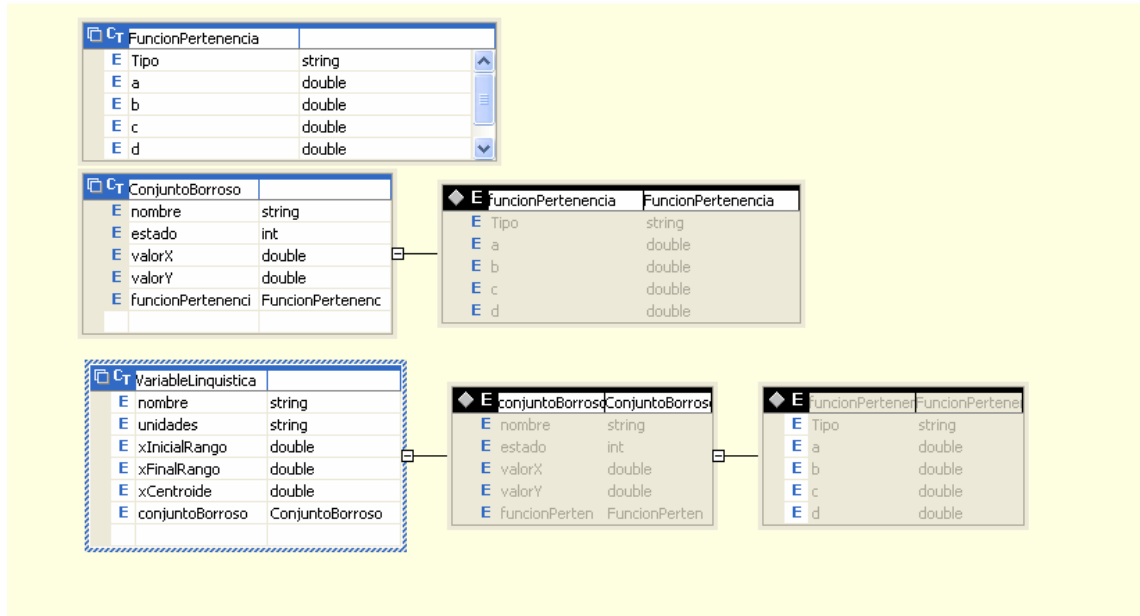


Figura 58. Serializacion Funciones de Pertenencia, Conjuntos Borrosos y Variables Lingüísticas.

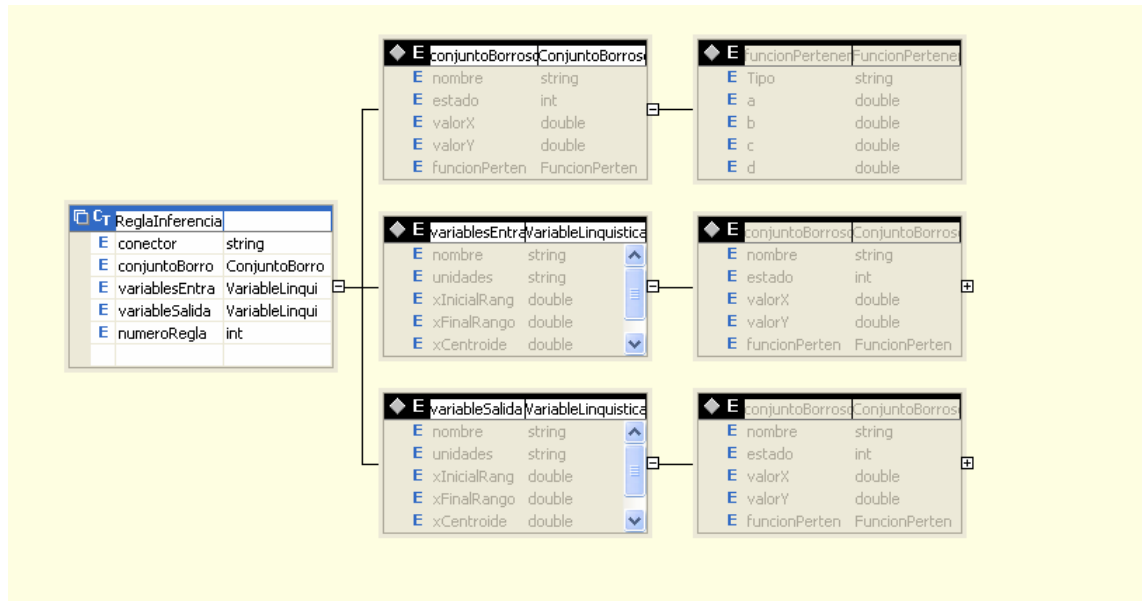


Figura 59. Serialización Reglas de Inferencia.

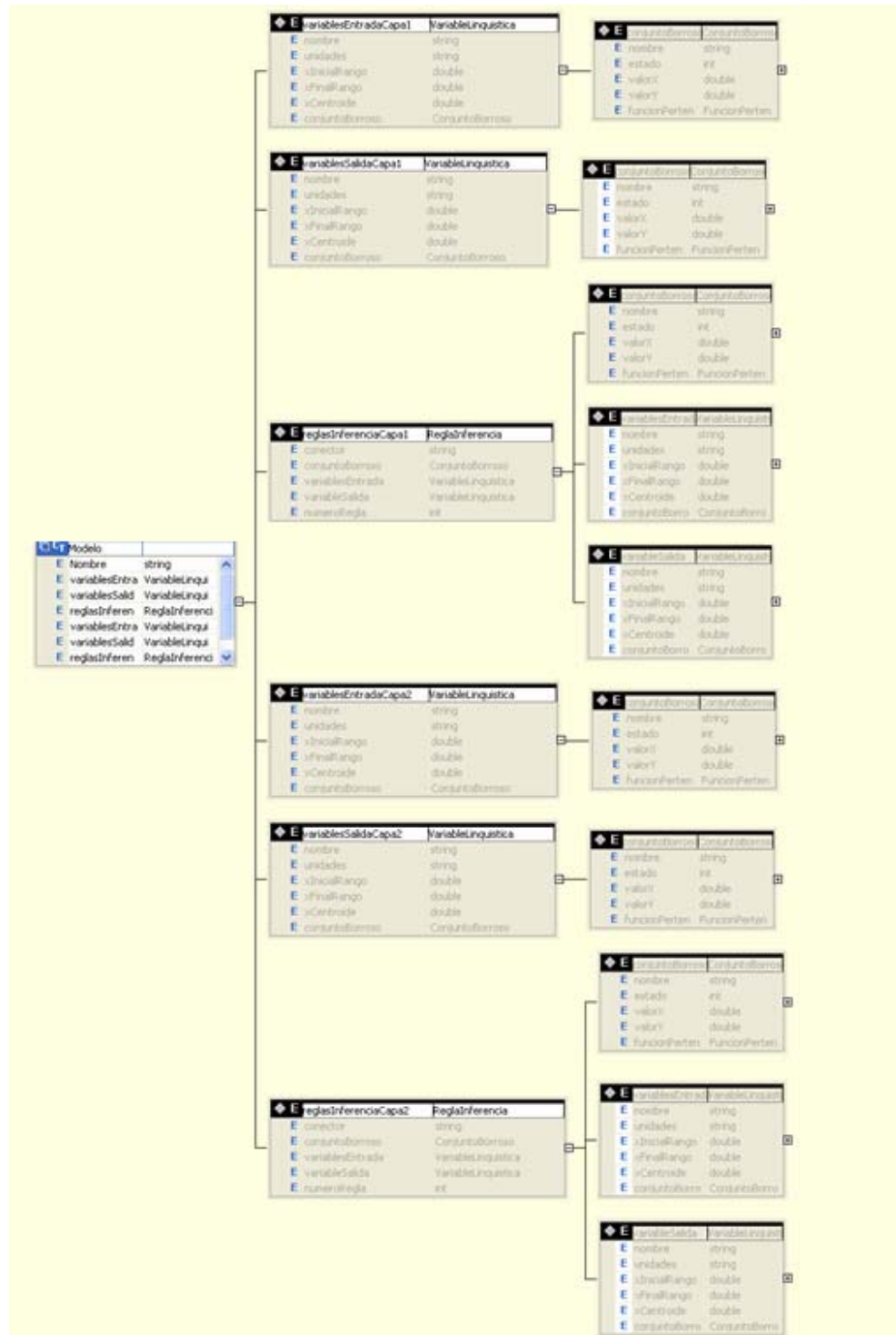


Figura 60. Serialización Modelo.

3.4. FASE DE TRANSICIÓN

Durante la fase de transición se busca comprobar el correcto funcionamiento de la aplicación, realizando una serie de pruebas que demuestren que cumple con todos los requisitos propuestos, además se realizó el manual de usuario que ofrece una ayuda para guiar al usuario en el manejo del prototipo.

3.4.1. ALCANCE DE LA FASE

En este punto se espera realizar las siguientes actividades:

Probar el prototipo a través de modelos difusos extraídos de artículos publicados en internet.

Verificar el cumplimiento de los requisitos propuestos para la aplicación.

Lectura y análisis de los artículos de los que se extrajeron los modelos difusos.

Elaboración del manual de usuario.

3.4.2. ESPECIFICACIÓN DE PRUEBAS

Se elaboran tres pruebas que se describen a continuación:

La primera consiste en plantear un modelo a partir del artículo de diseño de plantas descrito en el numeral 2.2 del capítulo 2, el cual se elabora en el prototipo y en Matlab para observar los valores obtenidos con ambas aplicaciones y concluir cómo es el desempeño del prototipo, teniendo en cuenta los resultados. Se escogió Matlab para realizar éstas pruebas debido a su reconocida experiencia en lógica difusa. Los resultados obtenidos en ésta fase se expondrán más adelante en el siguiente capítulo.

La segunda consiste en representar dos situaciones que utilicen sistemas difusos de dos capas, una es el diseño de plantas y la otra la determinación de picos en el flujo trans-mitral y trans-aórtico del corazón, para esto se elaboran los modelos que representan dichas situaciones en los que se establecen valores para las variables de entrada de modo que se muestren los resultados obtenidos.

La tercera consiste en estimar los tiempos de procesamiento del prototipo para determinar su eficiencia.

3.4.3. MANUAL DE USUARIO

En esta sección se explicaran las diferentes funcionalidades de la aplicación; la cual posee 8 ventanas, cada una de las cuales muestra los componentes que conforman un modelo difuso y dos alternativas de análisis de sensibilidad.

A continuación se esquematiza la distribución del prototipo.

DISTRIBUCIÓN DE MENÚS

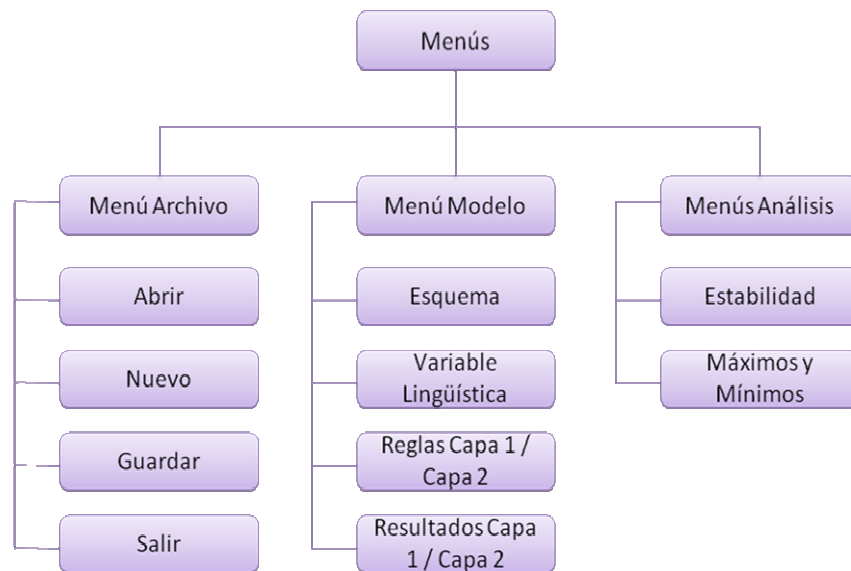


Figura 61. Distribución de Menús.

DISTRIBUCIÓN VENTANA ESQUEMA

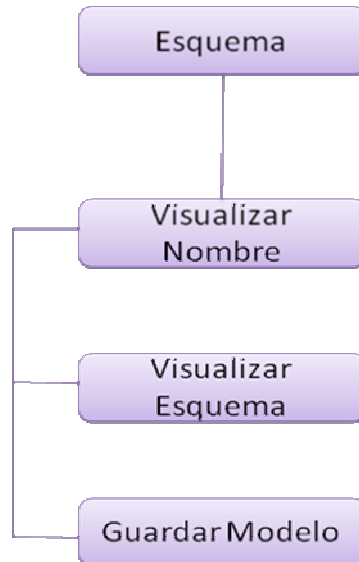


Figura 62. Ventana Esquema.

DISTRIBUCIÓN VENTANA VARIABLE LINGÜÍSTICA

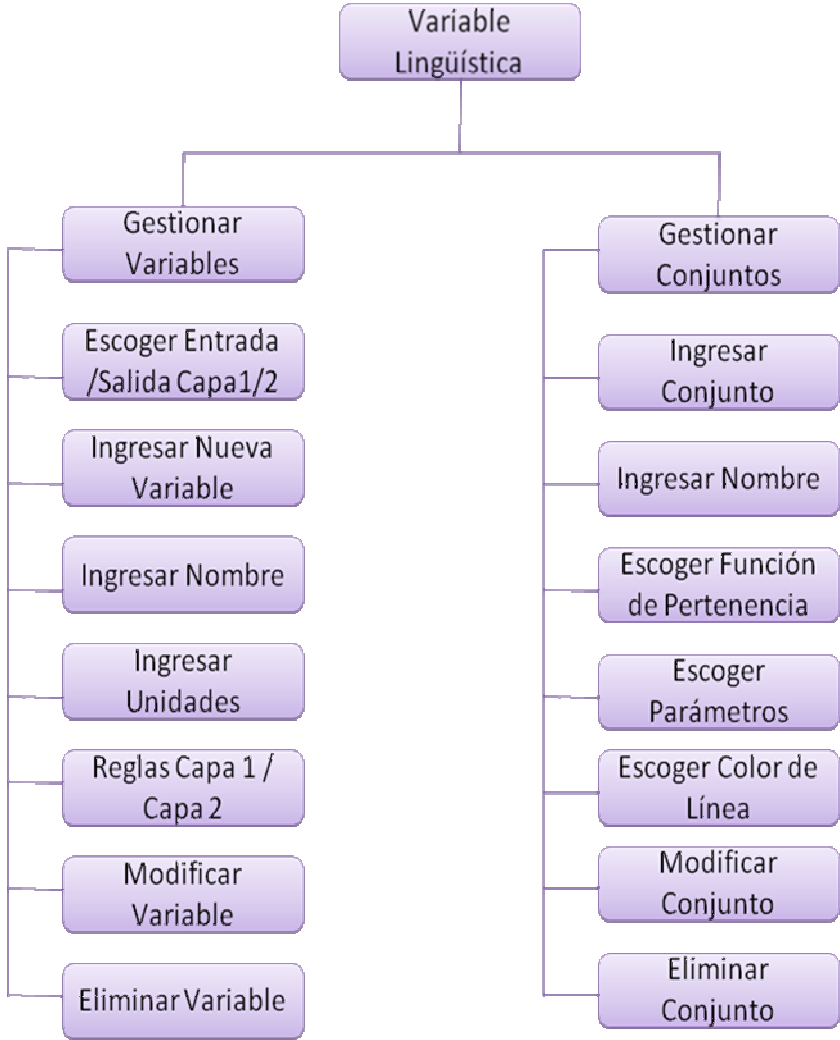


Figura 63. Ventana Variable Lingüística.

DISTRIBUCIÓN VENTANA REGLAS CAPA 1 Y CAPA 2

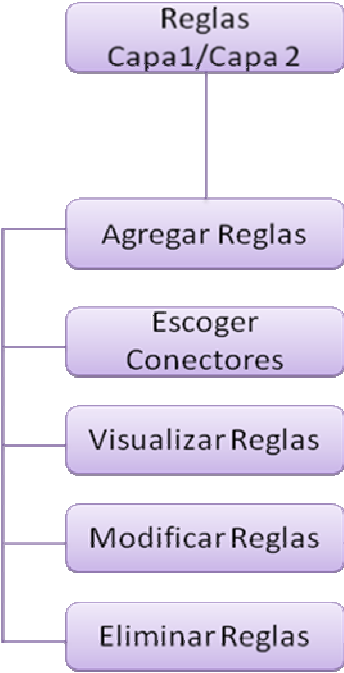


Figura 64. Ventana Reglas Capa 1 / Capa 2.

DISTRIBUCIÓN VENTANA RESULTADOS CAPA 1

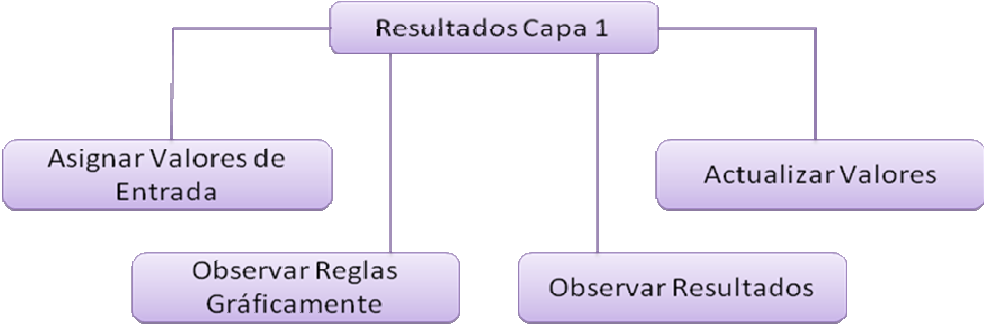


Figura 65. Ventana Resultados Capa1.

DISTRIBUCIÓN VENTANA CAPA 2

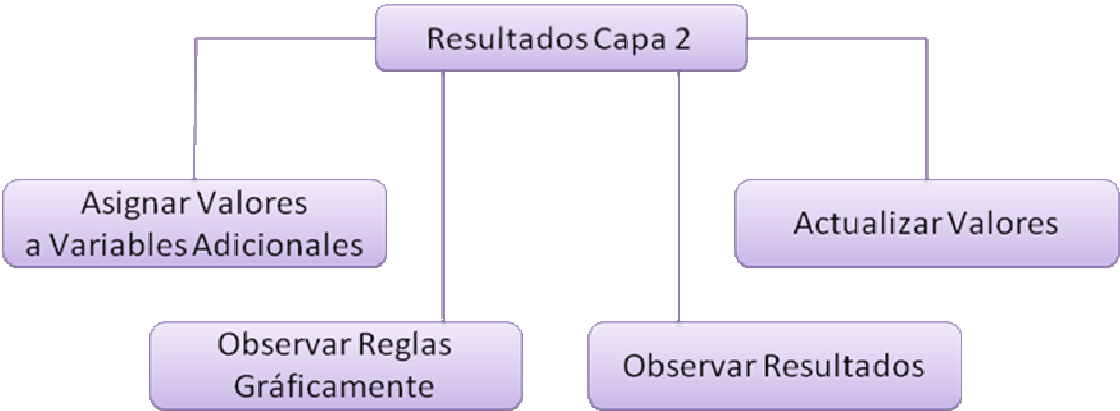


Figura 66. Ventana Resultados Capa2.

DISTRIBUCIÓN VENTANA ESTABILIDAD

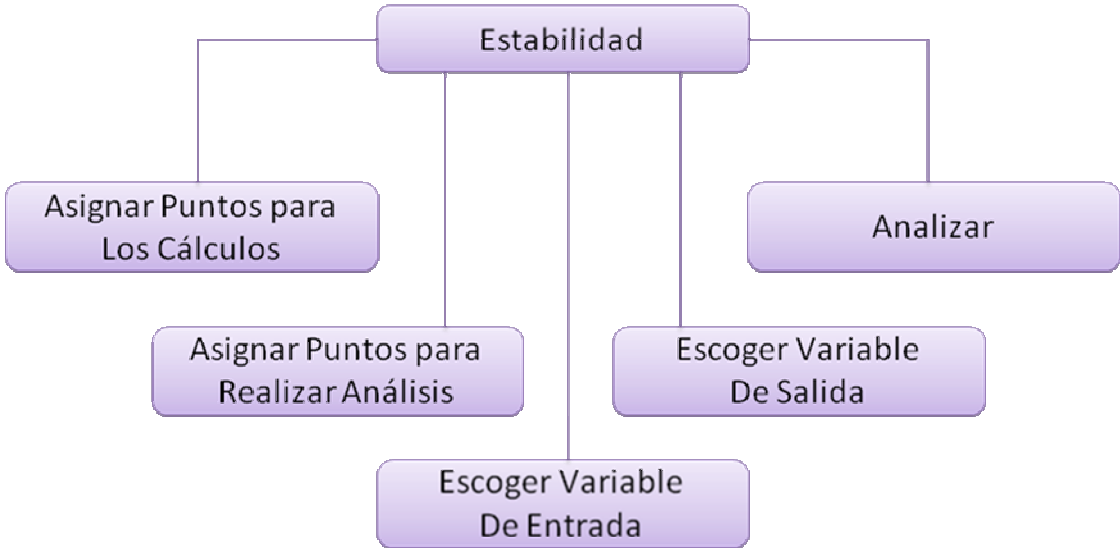


Figura 67. Ventana Estabilidad.

DISTRIBUCIÓN VENTANA MÁXIMOS Y MÍNIMOS

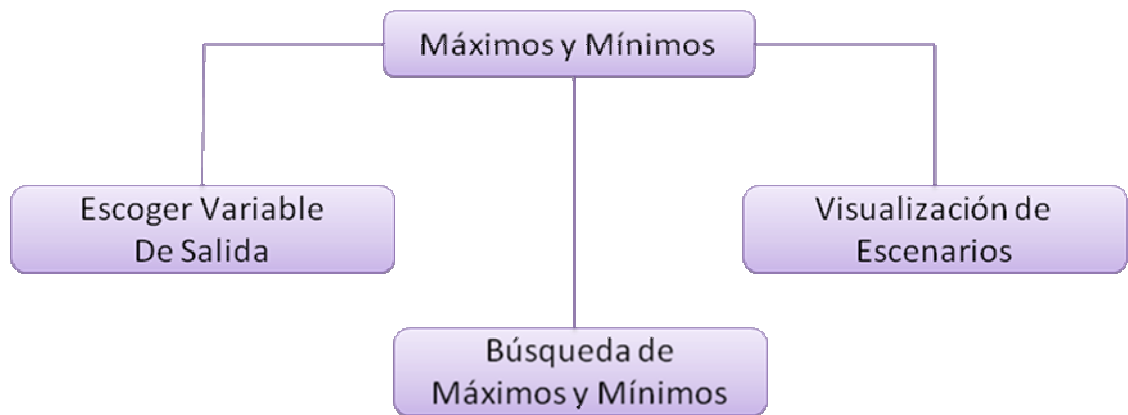


Figura 68. Ventana Máximos y Mínimos.

3.4.3.1. MENÚS

A continuación se presenta una descripción del prototipo, mostrando la funcionalidad de cada una de sus ventanas y sus componentes.

MENÚ ARCHIVO

En el menú archivo se encuentran cuatro submenús:

Nuevo: Crea nuevos archivos.

Abrir: Busca archivos que hayan sido almacenados con anterioridad.

Guardar: Almacena modelos, especificando la ruta donde se desean conservar.

Salir: Sale de la aplicación.

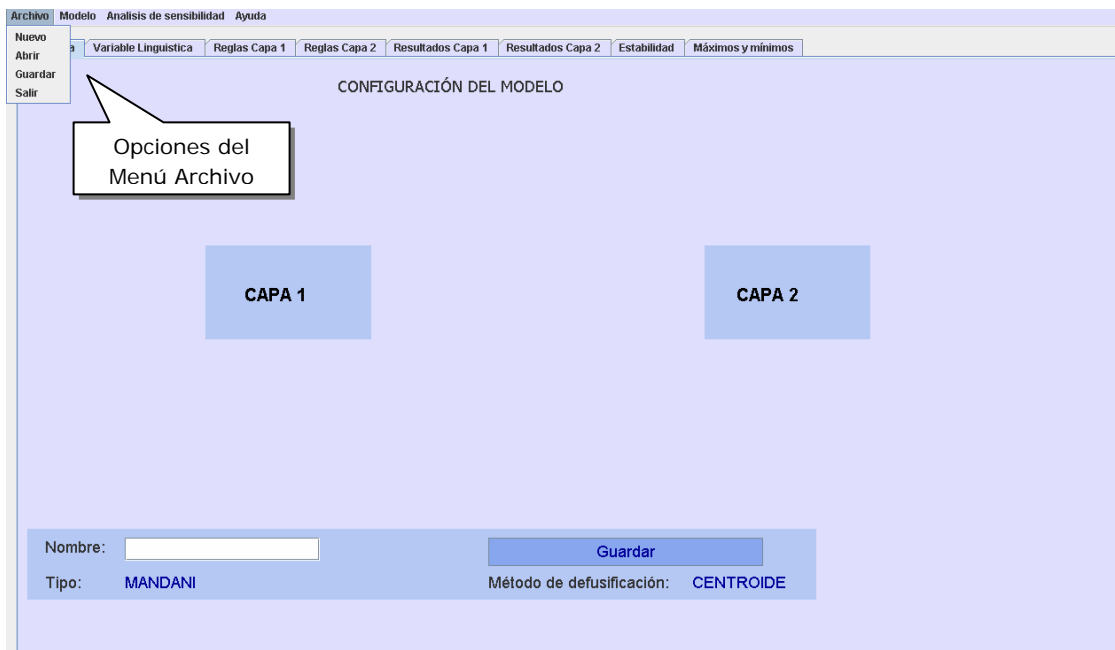


Figura 69. Menú Archivo.

MENÚ MODELO

El menú modelo se encarga de referenciar las siguientes ventanas: Esquema, Variables, Reglas Capa1, Reglas Capa2, Resultados Capa1 Y Resultados Capa2.

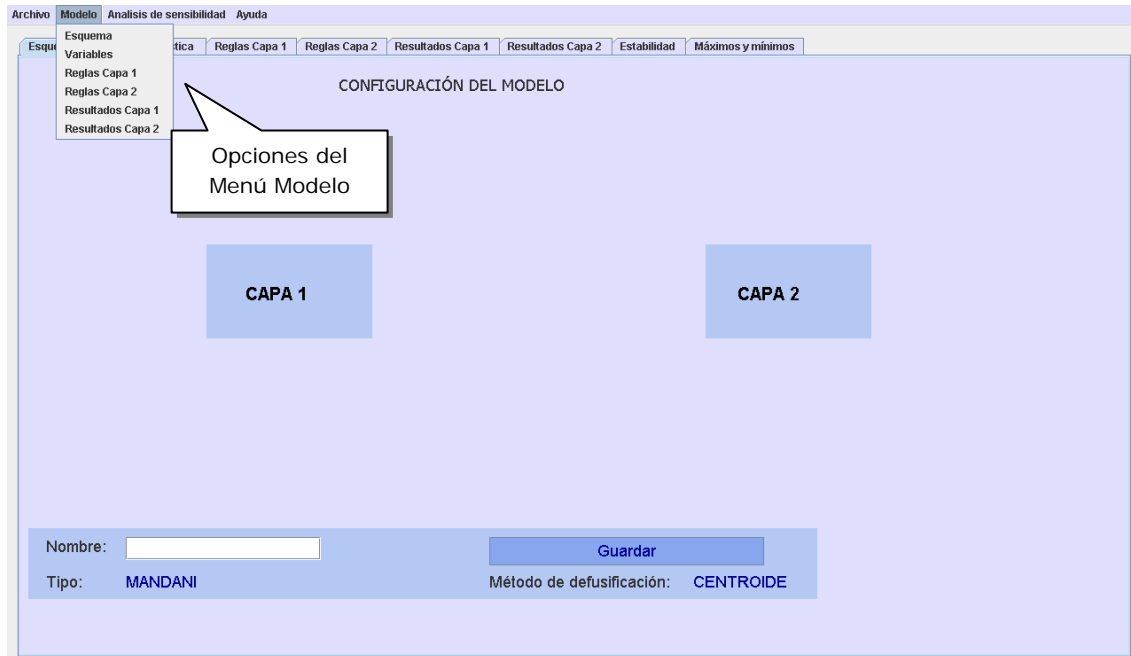


Figura 70. Menú Modelo.

MENÚ ANÁLISIS DE SENSIBILIDAD

El menú análisis de sensibilidad se encarga de referenciar las ventanas Estabilidad y Máximos y Mínimos.

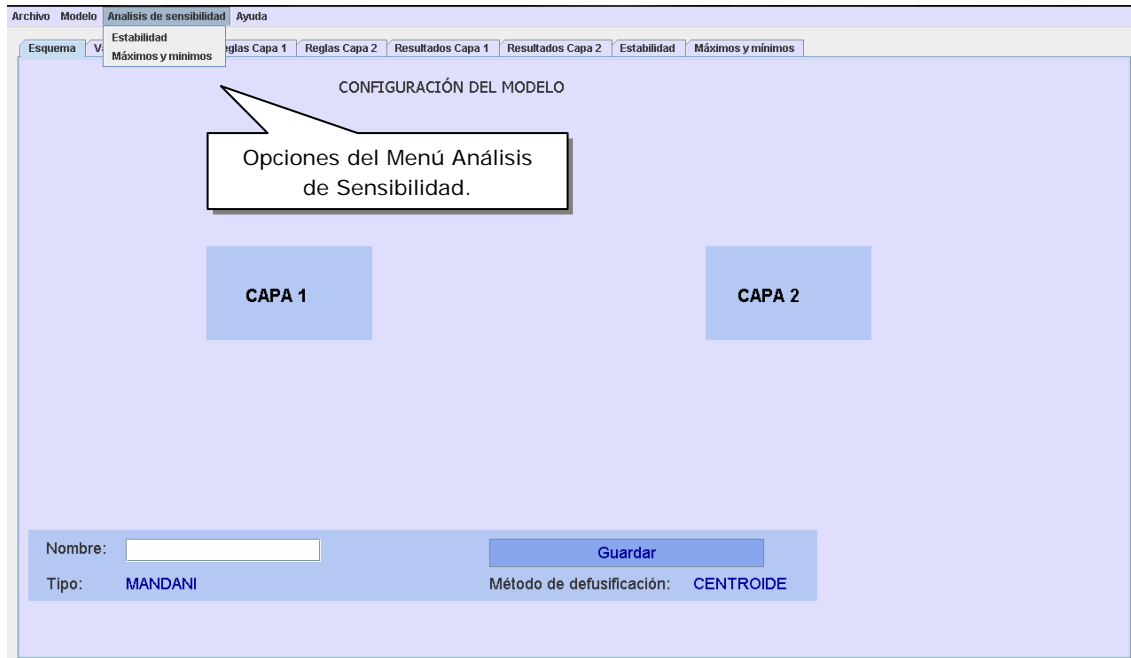


Figura 71. Menú Análisis de Sensibilidad.

3.4.3.2. VENTANAS

VENTANA ESQUEMA

En esta ventana se presenta el esquema general del modelo, mostrando las diferentes variables de entrada y salida correspondientes, el sistema de lógica difusa utilizado (Tipo Mamdani), el método de defusificación (centroide), el nombre y la opción de guardar el modelo, como se muestra en la figura 41.

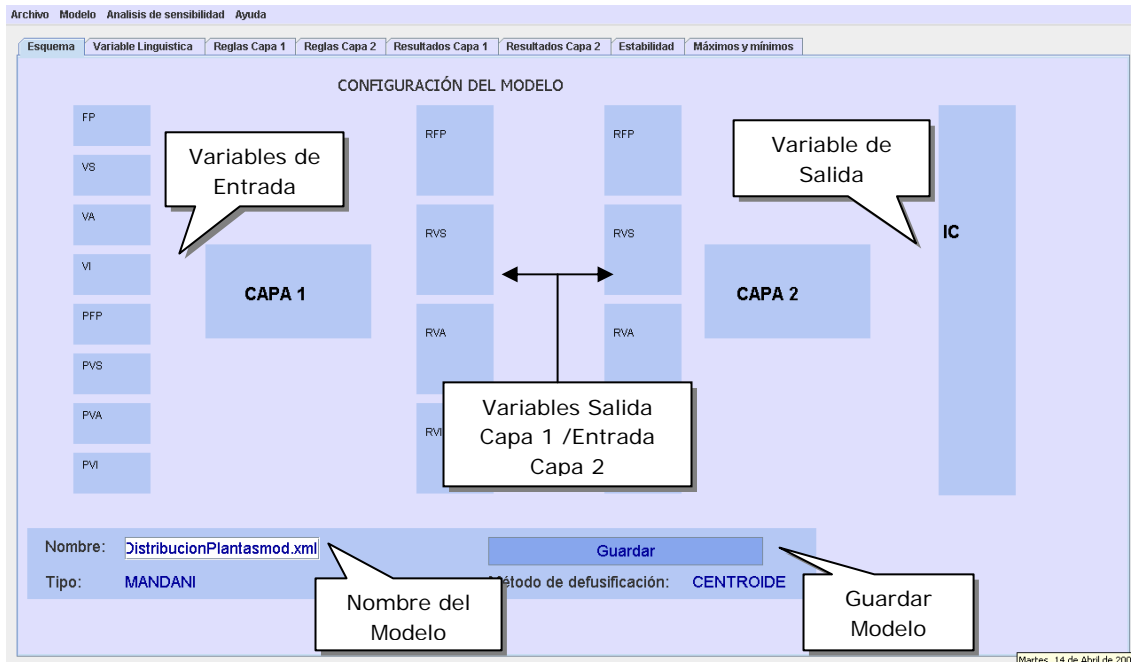


Figura 72. Ventana Esquema.

VENTANA VARIABLE LINGÜÍSTICA

Esta ventana se encuentra dividida en 2 secciones, una referente a la gestión de variables lingüísticas donde se escoge si son de entrada o salida de la capa 1 ó 2, nombre, unidades, y rango especificando el mínimo y el máximo; se hallan tres botones, Nueva Variable, Eliminar y Actualizar.

La otra sección comprende la gestión de los conjuntos difusos, allí se les asigna nombre, función de pertenencia, color de línea y parámetros asociados a las funciones; se encuentran tres botones, Nuevo Conjunto, Eliminar y Actualizar Parámetros.

Por último, en la parte superior de la ventana, se ubican las gráficas de las variables y los conjuntos creados.

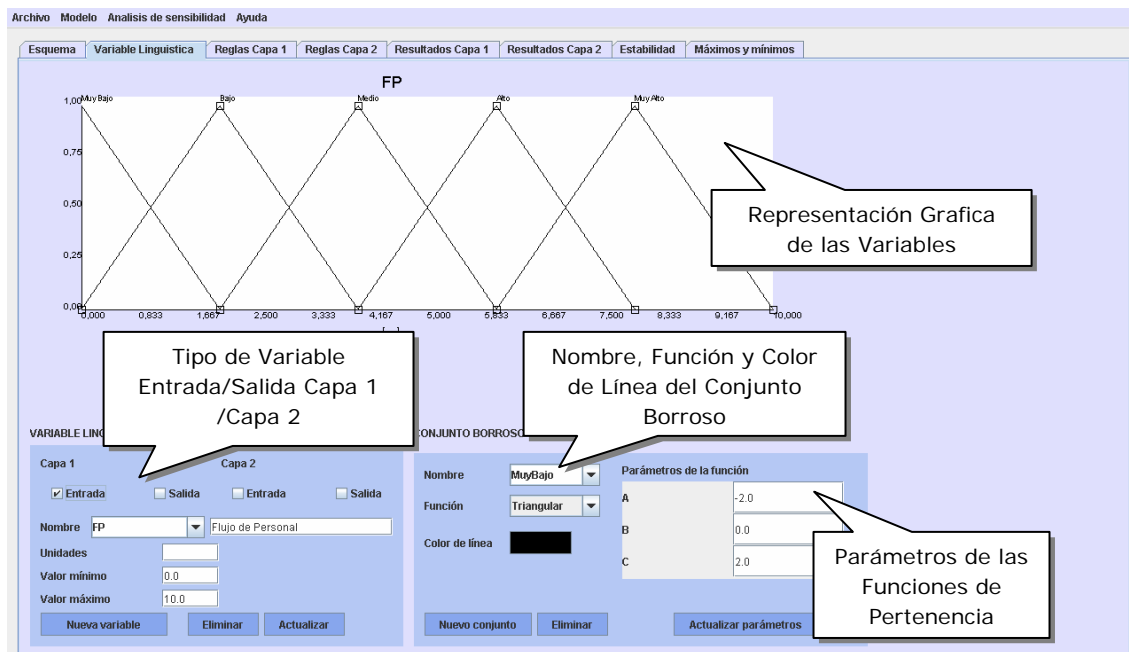


Figura 73. Ventana Variable Lingüística.

VENTANA REGLAS CAPA 1

Se divide en 4 partes, en la primera, se observan las reglas que se van creando, en la segunda, se encuentran las variables de entrada con sus respectivos conjuntos, en la tercera, están las variables de salida con sus conjuntos y finalmente aparecen los conectores “y” u “o”, a través de los cuales se relacionan los conjuntos difusos, junto con los botones borrar y agregar.

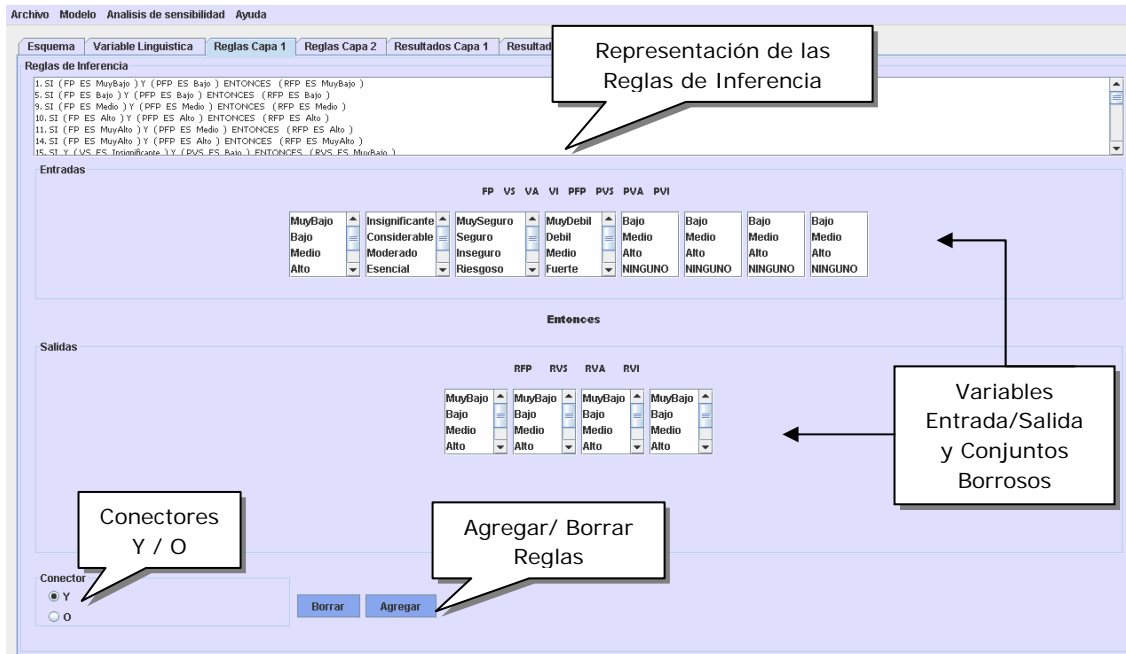


Figura 74. Ventana Reglas Capa 1.

VENTANA REGLAS CAPA 2

Teniendo en cuenta que las funciones y organización de ésta ventana son iguales a la anterior, el usuario puede referirse allí para conocer su funcionamiento.

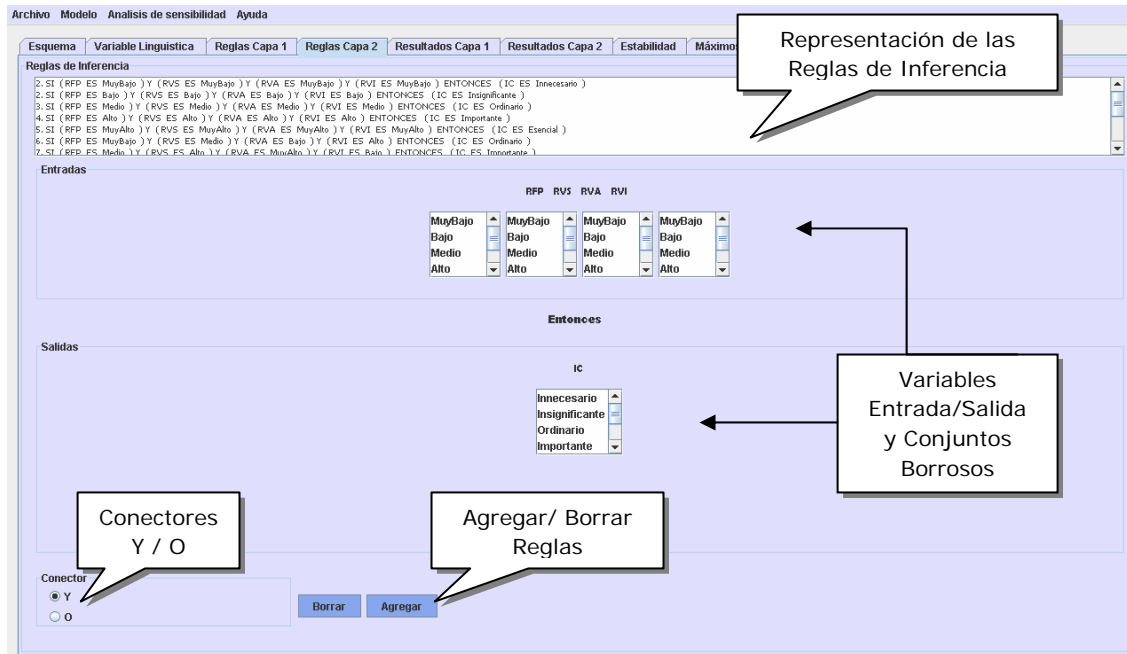


Figura 75. Ventana Reglas Capa 2.

VENTANA RESULTADOS CAPA 1

En ésta ventana se observan las reglas de forma gráfica ubicadas en filas y en el orden que fueron creadas, cada columna simboliza una variable de entrada o salida, diferenciándose ésta última por poseer una fila adicional en la parte inferior que representa su resultado; debajo del nombre de cada variable aparece el valor numérico que toma, el cual puede ser modificado a través de un indicador representado por una línea roja que se desplaza a lo largo de su columna. Si se desea asignar el valor a una variable manualmente, en la parte inferior de la ventana se ubican cajas de texto asociadas a ellas para realizar ésta acción seguida de la pulsación del botón Actualizar.

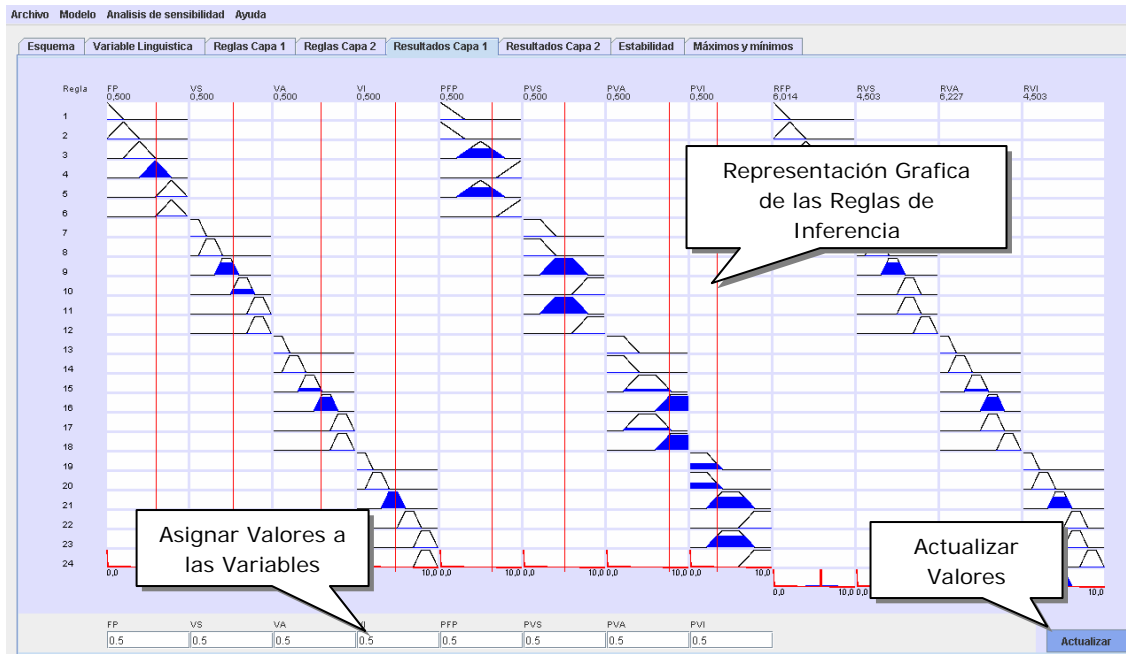


Figura 76. Ventana Resultados Capa 1.

VENTANA RESULTADOS CAPA 2

En ésta ventana se observan las reglas de forma gráfica ubicadas en filas y en el orden que fueron creadas, cada columna simboliza una variable de entrada o salida, diferenciándose ésta última por poseer una fila adicional en la parte inferior que representa su resultado; en ésta ventana no es posible modificar los valores de las variables de entrada cuando representan salidas de la capa 1 puesto que éstos valores dependen de los resultados obtenidos. Si el modelo contiene variables adicionales de entrada en la capa 2, los valores asociados pueden ser introducidos de la misma manera que en la ventana resultados capa 1.

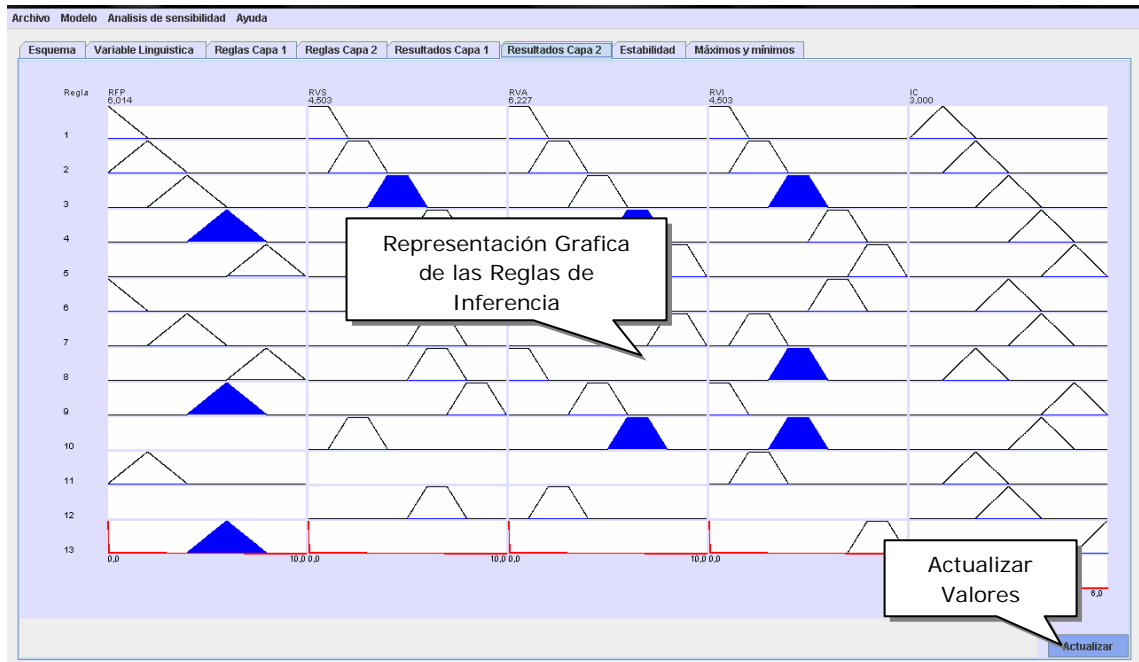


Figura 77. Ventana Resultados Capa 2.

VENTANA ESTABILIDAD

En ésta ventana, se aprecia el número de reglas sugerido y el número de reglas actual para cada una de las capas, fijar los puntos para realizar los cálculos y los puntos a analizar teniendo en cuenta que a mayor cantidad de puntos, el resultado se aproximará mejor a la realidad, cabe resaltar que en este caso, el tiempo de respuesta se incrementa. El usuario escoge la variable de entrada y salida sobre las cuales desea realizar el proceso. Finalmente, se muestran los escenarios, en cada uno se presentan los valores de entrada y "Delta Max" el cual representa el cambio máximo que se registra en los valores de la salida junto con la grafica de las variables escogidas.

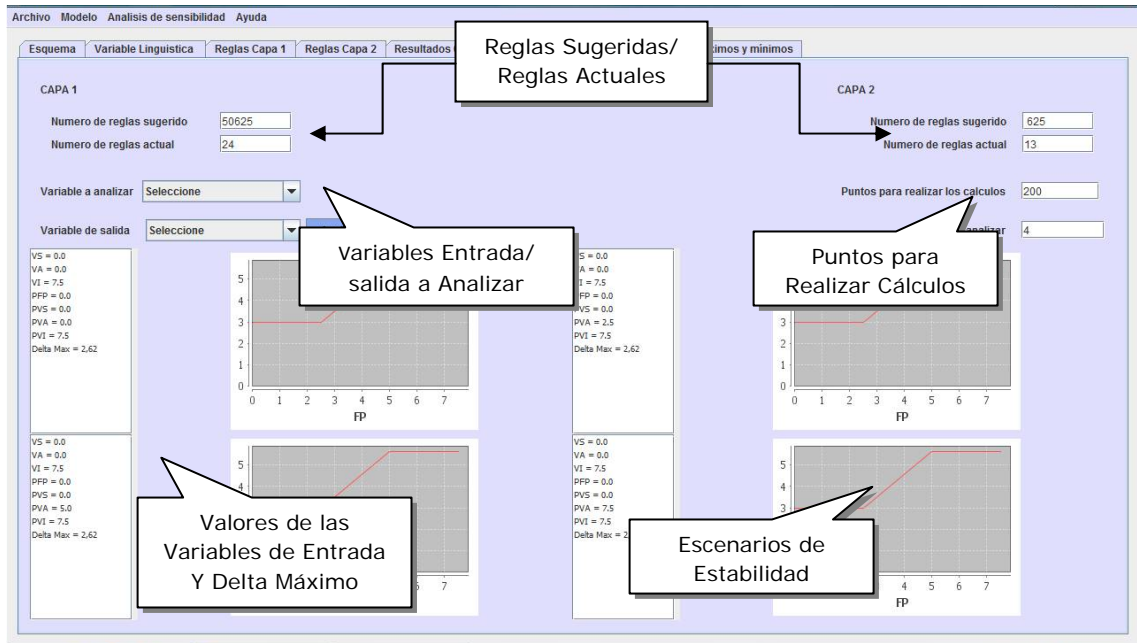


Figura 78. Ventana Estabilidad.

VENTANA MÁXIMOS Y MÍNIMOS

En esta ventana se escoge la variable de salida para realizar la búsqueda de los puntos máximos y mínimos, en este momento se generan 20 escenarios que presentan los 10 mayores máximos y los 10 menores mínimos y los valores asociados a cada una de las variables de entrada que los generan.

CAPÍTULO IV

4. PRUEBAS Y RESULTADOS

A continuación se muestra el desarrollo de las pruebas descritas en el numeral 3.4.2 del capítulo 3 del presente documento.

4.1. COMPARACIÓN DE RESULTADOS ENTRE EL PROTOTIPO Y MATLAB

Ésta comparación se realizó planteando un modelo difuso de dos capas fundamentado en el artículo²⁵ de diseño de plantas publicado en internet, se tomaron 4 valores aleatorios para cada variable de entrada y se registraron los valores de salida del proceso, con los que se determinó el error relativo porcentual existente entre los resultados obtenidos por MATLAB y los resultados del prototipo para concluir la manera como se desempeña este último.

Las variables de entrada que se presentan en el modelo son:

FP: Flujo de Personal

VS: Vinculo de Supervisión

VA: Vinculo Ambiental

VI: Vinculo de Información

PFP: Peso del Flujo de Personal

PVS: Peso del Vinculo de Supervisión

PVA: Peso del Vinculo Ambiental

²⁵ “Facilities layout planning based on Fuzzy multiple criteria decision-making methodology”

PVI: Peso del Vinculo de Información

De la misma manera las variables de salida de la capa 1 y entrada de la capa 2 son:

RFP: Resultante Flujo de Personal

RVS: Resultante Vinculo de Supervisión

RVA: Resultante Vinculo Ambiental

RVI: Resultante Vinculo de Información

Por último la salida de la capa 2 es IC o Índice de cercanía que es el resultado según el cual se determina el error para realizar la comparación.

Los valores asumidos para las variables anteriormente mencionadas junto con sus resultados se muestran a continuación:

		FP	VS	VA	VI	PFP	PVS	PVA	PVI	RFP	RVS	RVA	RVI	IC
FUSYSTWO	Prueba1	0.5	0.3	9.1	0.25	0.8	0.7	8.5	2.52	1.512	0.781	8.503	0.832	1.064
	Prueba2	0.74	1.59	7.53	3.27	2.12	3.62	6.78	2.66	2.003	2.003	7.326	3.033	2.161
	Prueba3	3.35	1.97	5.13	4.97	4.167	0.95	3.85	5.75	4.318	2.483	4.503	4.503	4.196
	Prueba4	8.35	6.75	2.86	7.98	9.25	8.52	0.97	6.85	8.350	6.503	2.503	7.329	5.579
MATLAB	Prueba1	0.5	0.3	9.1	0.25	0.8	0.7	8.5	2.52	1.38	0.752	8.5	0.803	1.66
	Prueba2	0.74	1.59	7.53	3.27	2.12	3.62	6.78	2.66	1.83	1.93	7.5	3.11	2
	Prueba3	3.35	1.97	5.13	4.97	4.167	0.95	3.85	5.75	4	2.47	4.5	4.5	4
	Prueba4	8.35	6.75	2.86	7.98	9.25	8.52	0.97	6.85	8	6.5	2.5	8.41	5.69

Tabla 11. Datos Utilizados en la Comparación de Resultados.

IC Prototipo	IC Matlab	% Error
1.064	1.66	35.90
2.161	2	8.05
4.196	4	4.9
5.579	5.69	1.95

Tabla 12. Resultados del Porcentaje Error Relativo

La tabla 12 muestra los porcentajes de error obtenidos de la comparación en cada una de las cuatro pruebas realizadas, la mayor parte de estos son mínimos, lo cual indica que el prototipo se aproxima a los valores esperados en el planteamiento del modelo, los altos porcentajes de error que se presenten pueden deberse a los métodos utilizados en el proceso de defusificación.

4.2.APLICACIÓN DE LOS MODELOS DE DISEÑO DE PLANTAS Y DETERMINACIÓN DE PICOS EN EL FLUJO TRANS-MITRAL Y TRANS-AÓRTICO DEL CORAZÓN.

4.2.1. DISEÑO DE PLANTAS:

La localización de las instalaciones en las organizaciones son importantes ya que permiten agilizar procesos como los de manufactura o calidad, estas localizaciones requieren estar bajo condiciones favorables como fuentes de materias primas, mano de obra y demandas en el mercado que contribuyan al éxito de los procesos.

Ningún procedimiento de localización puede asegurar que se ha escogido el mejor, pero es importante tener en cuenta el rendimiento que una planta requiere en función de la ubicación de sus componentes con el fin de minimizar costos, mejorar su funcionamiento y aumentar su producción.

Una embotelladora de gaseosas realiza un estudio para determinar el índice de cercanía existente entre dos facilidades, los equipos dosificadores²⁶ y la máquina de llenado²⁷, se tienen en cuenta factores como el flujo de personal, el vínculo de

²⁶ En una embotelladora de gaseosas el equipo dosificador combina los ingredientes del producto.

²⁷ Introducción del líquido en el envase.

supervisión, el vínculo ambiental y el vínculo de información a los que se les asignan valores en una escala de 0 a 10 y los pesos atribuidos a cada factor.

El flujo de personal y el peso establecidos entre los equipos dosificadores y la máquina de llenado deben ser Altos ya que se requiere una cantidad importante de individuos controlando los procesos de las máquinas lo cual representa un factor importante en la relación de las mismas.

El vínculo de supervisión entre las facilidades debe ser moderado y el peso medio ya que la labor del supervisor asignado a estas áreas es importante pero no relevante en relación a las demás.

El vínculo ambiental entre las facilidades debe ser seguro y el peso bajo ya que los materiales manipulados y el entorno no representan factores de peligro que afecten las actividades que se realizan.

El vínculo de información entre las facilidades debe ser fuerte y el peso alto ya que la función que cumple el equipo dosificador determina el proceso que la máquina de llenado realiza.

Variables	Valores asignados
FP	7.5
VS	4.57
VA	2.23
VI	6
PFP	10
PVS	7.94

PVA	0.51
PVI	8.49
IC	4.026

Tabla 13. Valores de las variables del Modelo Diseño de Plantas.

El índice de cercanía encontrado con el prototipo fue de 4.026, representa que la cercanía existente entre los equipos dosificadores y la máquina de llenado es importante por lo tanto la distancia entre ellos debe ser corta aunque no deben encontrarse demasiado cerca. Esto da a los expertos una idea de la distancia que debe haber entre las facilidades.

4.2.2. DETERMINACIÓN DE PICOS EN EL FLUJO TRANS-MITRAL Y TRANS-AÓRTICO DEL CORAZÓN

Los electrocardiogramas y espectrogramas son herramientas que facilitan a los especialistas en cardiología diagnosticar enfermedades del corazón, analizando los distintos picos que se presentan en ellos.

Un modelo fuzzy elaborado para determinar la probabilidad que tiene un pico, en un ECG y en un espectrograma, de ser un pico que represente la abertura o cierre de la válvula trans-mitral, permite encontrar éstos picos de forma automática facilitando la labor de los expertos. Lo mismo ocurre con un modelo fuzzy que encuentra la probabilidad de un pico de representar la abertura o cierre de la válvula aortica.

Al analizar el flujo de sangre a través de la válvula mitral por medio de un ECG o espectrograma:

Si un pico tiene una amplitud positiva baja, una pendiente positiva baja y un tiempo de duración medio entonces la probabilidad de que éste pico represente el cierre de la válvula es alta.

Si un pico tiene una amplitud positiva baja, una pendiente positiva baja y un tiempo de duración bajo entonces la probabilidad de que éste pico represente el cierre de la válvula es media.

Si un pico tiene una amplitud negativa alta, una pendiente positiva alta y un tiempo de duración medio entonces la probabilidad de que éste pico represente la abertura de la válvula es baja.

Si un pico tiene una amplitud negativa baja, una pendiente negativa baja y un tiempo de duración bajo entonces la probabilidad de que éste pico represente la abertura de la válvula es alta.

Amplitud	Pendiente	Duración	PP_AperturaE	PP_CierreE	PP_AperturaBpc	PP_CierreBpc
2.35	45	0.093	0.064	0.936	0.064	0.936
2.35	45	0.033	0.935	0.5	0.935	0.5
-15	90	0.076	0.194	0.806	0.194	0.806
-15	90	0.025	0.806	0.5	0.806	0.5

Tabla 14. Valores de las variables del Modelo Determinación de Picos de los Flujos Trans-Mitral.

Al analizar el flujo de sangre a través de la válvula sigmoidea aortica por medio de un espectrograma o ECG:

Si un pico tiene una amplitud positiva alta, una pendiente negativa baja y un tiempo de duración bajo entonces la probabilidad de que éste pico represente la abertura de la válvula es alta.

Si un pico tiene una amplitud positiva baja, una pendiente positiva baja y un tiempo de duración alto entonces la probabilidad de que éste pico represente el cierre de la válvula es alta.

Si un pico tiene una amplitud positiva alta, una pendiente negativa baja y un tiempo de duración alto entonces la probabilidad de que éste pico represente la abertura de la válvula es baja.

Si un pico tiene una amplitud positiva baja, una pendiente positiva baja y un tiempo de duración bajo entonces la probabilidad de que éste pico represente el cierre de la válvula es baja.

Amplitud	Pendiente	Duración	PP_AberturaE	PP_CierreE	PP_AberturaBpc	PP_CierreBpc
18	-37	0.036	0.823	0.177	0.823	0.177
2.25	56	0.22	0.064	0.936	0.064	0.936
18	-37	0.22	0.177	0.823	0.177	0.823
2.25	56	0.036	0.933	0.068	0.933	0.068

Tabla 15. Valores de las variables del Modelo Determinación de Picos de los Flujos Trans-Aórticos.

Como se observa en las tablas “x” y “y” la duración es un factor determinante en la búsqueda de picos de abertura o cierre.

Las probabilidades encontradas definen cuando un pico tiende más a ser de abertura o cierre.

4.3. ESTIMACIÓN DE LOS TIEMPOS DE PROCESAMIENTO DEL PROTOTIPO

Para estimar la eficiencia de procesamiento del prototipo se realizaron 6 pruebas para analizar la ventana estabilidad y 3 pruebas para la ventana máximos y mínimos, se tomaron los mismos valores de las variables de entrada utilizados en el numeral anterior y se registraron los tiempos durante los cuales se ejecuto el modelo. Para el desarrollo de estas pruebas se utilizo el ejemplo de diseño de plantas y se utilizo como variable de entrada el Flujo de Personal y el Índice de Cercanía como variable de salida.

La siguiente tabla presenta los tiempos de procesamiento obtenidos para la ventana de estabilidad teniendo en cuenta el número de puntos para realizar los cálculos y el número de puntos a analizar.

PUNTOS PARA REALIZAR CÁLCULOS	PUNTOS A ANALIZAR	TIEMPO DE LA PRUEBA
100	3	1 Minuto
200	3	1 Minuto
100	4	4 Minutos
200	4	8 Minutos
100	5	25 Minutos

200	5	43 Minutos
-----	---	------------

Tabla 16. Tiempos de Procesamiento de la Ventana Estabilidad.

La siguiente tabla presenta los tiempos de procesamiento obtenidos para la ventana de máximos y mínimos teniendo en cuenta el número de puntos para realizar los cálculos y la variable de salida seleccionada.

PUNTOS PARA REALIZAR LOS CÁLCULOS	TIEMPO DE LA PRUEBA
50	9 Minutos
100	22 Minutos
150	30 Minutos

Tabla 17. Tiempos de Procesamiento de la Ventana Máximos y Mínimos.

Se concluye que a mayor cantidad de puntos para realizar los cálculos, mayor es el tiempo que demora la aplicación en dar un resultado, cabe resaltar que a medida que se incrementa el valor de dichos puntos se obtendrá un resultado más preciso.

CAPÍTULO V

5. CONCLUSIONES

FusysTwo brinda soporte al proceso de toma de decisiones y solución de problemas haciendo uso de la lógica difusa.

FusysTwo contribuye a los estudiantes como herramienta de trabajo en materias relacionadas, permitiendo la visualización de las etapas de creación y solución de un modelo difuso.

La herramienta, logra que los usuarios extiendan sus conocimientos y reconozcan la importancia de utilizar la lógica difusa y la manera como se constituye.

El manual de usuario es una guía práctica en la creación de modelos y en el funcionamiento de la aplicación.

Con la persistencia de datos en xml se logra un buen desempeño en el almacenamiento y recuperación de datos; y consigue que los archivos creados sean portables.

El desarrollo de este proyecto constituyó un aporte significativo para la formación profesional de sus autoras, aplicando los conocimientos adquiridos durante su formación universitaria y apropiando nuevos.

Los componentes del análisis de sensibilidad; estabilidad, máximos y mínimos constituyen un recurso importante para obtener conclusiones sobre el comportamiento de un modelo.

El tiempo de procesamiento para el análisis de sensibilidad se incrementa en la medida en que existan más variables de entrada y dependiendo de los puntos para realizar cálculos y los puntos a analizar

CAPÍTULO VI

6. RECOMENDACIONES Y TRABAJOS FUTUROS

Mejorar la efectividad del algoritmo de análisis de sensibilidad contemplando técnicas alternativas para reducir su tiempo de procesamiento.

Implementar un método que permita encontrar las variaciones de una o más variables de salida con respecto a cada una de las variables de entrada asociadas a un modelo dentro de un mismo proceso.

Se recomienda la utilización del manual de usuario para garantizar el adecuado manejo de la aplicación en la solución de modelos de una y dos capas.

Es importante el uso de la aplicación por parte de otros usuarios para identificar posibles falencias y contribuir en la elaboración de una nueva versión.

CAPÍTULO VII

7. BIBLIOGRAFÍA

Durante la investigación se han consultado fuentes de información de donde se ha extraído y recopilado información relevante y necesaria para el desarrollo del proyecto.

7.1. TEORÍA LÓGICA DIFUSA

1. KLIR, George J., YUAN, Bo. Fuzzy sets and fuzzy logic. Theory and applications. Upper Saddle River, NJ (USA). PRENTICE HALL PTR. 1995. 574p.
2. MARTINEZ QUINTERO, Jaime A., SANCHEZ VEGA, Bruce Lehmann. Lógica fuzzy. Fundamentación teórica y aplicaciones. Bucaramanga, Colombia. 2002. 269p.
3. Introducción Y Conceptos Básicos. Tipos De Sistemas De Lógica Difusa. Jesús Alfonso López. 2003. Disponible en Internet:
http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html
4. Capítulo 2: Descripción General de las técnicas de Lógica Difusa. Consultado el 19 Octubre, 2008. Disponible en Internet en :
http://www.tdr.cesca.es/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf

7.2. TEORÍA METODOLOGÍA PROCESO UNIFICADO DE DESARROLLO SE SOFTWARE Y DIAGRAMAS UML

1. Larman C. UML y Patrones. Introducción al análisis y Diseño Orientado a Objetos. Pearson.
2. Piattini M, Análisis y Diseño de aplicaciones Informáticas de Gestión: Una perspectiva de Ingeniería del Software. México: Alfaomega; Madrid: Ra-Ma 2004.
3. Jacobson I, Booch G, Rumbaugh J. El proceso Unificado de Desarrollo de Software. Madrid: Pearson Educación S.A
4. Castro Robin A. Estructura básica del Proceso Unificado de Desarrollo de Software. Disponible en internet en:
http://dspace.icesi.edu.co/dspace/bitstream/item/399/1/rcastro_estructura-bas-puds.pdf.
5. Introducción al Proceso de Desarrollo de Software. Disponible en internet en:
http://www.webbress.com.ar/web/sitios/web_proyectoisi/download/proyecto_2005_b.pdf.

7.3. JAVA ,ARCHIVOS XML Y ANÁLISIS DE SENSIBILIDAD

1. Lenguaje de Programación Java. Disponible en internet en:
http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_Java
http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java

2. Serialización: Persistencia de Datos en Java. Disponible en internet en:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/montero_g_g/capitulo2.pdf.
3. Análisis Post Optimal y Algoritmo de Ramificación y Acotamiento.
 Disponible en Internet en:
<http://www.uv.es/asepuma/V/8.pdf>.
4. Optimización Difusa. Disponible en internet en:
http://www2.ing.puc.cl/gescopp/Sergio_Maturana/FORMULACIONGENERALL.html

7.4. *DISEÑO DE PLANTAS*

1. Fabricación de Bebidas Gaseosas. Disponible en Internet en:
http://www.anber.cl/inicio/variedad_prod_gaseosas.php.
2. GRANT IRESON, William. Planeamiento de fábricas. Barcelona, España. EDITORIAL HISPANO EUROPEA. 1966. Título de la edición original: "Factory planning and plant layout". 452p. Traducción por Francisco Javier Valls.
3. MUTHER, Richard. Distribución en planta. Barcelona, España. EDITORIAL HISPANO EUROPEA. 1965. Título de la obra original: "Practical plant layout". Traducción por Carmelo M^a Cabré.
4. S. K. Deby and B. Bhattacharyya. Production Engineering Department, Jadavpur University, Kolkata, India. Vol.41, no.18, 4487–4504. 2003.
<http://www.tandf.co.uk/journals>.

7.5. FLUJO TRANS- VALVULAR

1. El Corazón tiene dos Movimientos. Consultado el 10 Marzo, 2009.
http://www.proyectosalohogar.com/Diversos_Temas/Funcionamiento_Corazon.html
2. Corazón. Wikimedia Foundation, Inc. Consultado el 12 de Marzo, 2009.
<http://es.wikipedia.org/wiki/Coraz%C3%B3n>
3. Electrocardiograma. Wikimedia Foundation, Inc. Consultado el 15 de Marzo, 2009.
<http://es.wikipedia.org/wiki/Electrocardiograma>
4. Investigation of a Hierarchical Fuzzy Model for Echocardiographic Assessment of Trans-valvular Hemodynamic Indexes. Jie Gong, Richard Kirsner, Andrew Maclsaac, Carol Drossos, James Cameron. Department of Electronic Engineering, La Trobe University, Bundoora. Cardiac Investigation Unit, St Vincent's Hospital, Melbourne, Australia. Enero 2002.
http://www.aprs.org.au/dicta2002/dicta2002_proceedings/Gong13.pdf