

Universidad Central “Marta Abreu” de Las Villas
Facultad Matemática Física y Computación
Licenciatura en Ciencia de la Computación



Trabajo de Diploma

*«Sistema inteligente difuso para la evaluación de riesgos
tecnológicos»*

Autor: Anabel Martínez Machado

Tutores: Dra. María Matilde García Lorenzo

Dr. Abel Goya Valdivia

“Año 56 de la Revolución”

Santa Clara

2014

DECLARACIÓN JURADA



El que suscribe, _____, hago constar que el trabajo titulado _____ fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de _____, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Laboratorio

Fecha

DEDICATORIA

A Laurita, la lucecita de mi vida y la razón por la cual demoré en graduarme.

AGRADECIMIENTOS

En primer lugar a toda mi familia que me apoyó incondicionalmente para superar las dificultades, en especial a mi mamá y a mi papá que cuidaron de mi niña para que pudiera estudiar.

A mi novio maravilloso, Denis, mi mejor profesor y mi ejemplo a seguir.

A todos los profesores que me impartieron clases a lo largo de la carrera, por los conocimientos, la educación y sus aportes a mi formación como profesional, en especial a mi tutora Marilyn por su paciencia y exigencia y a mi tutor Abel por su guía constante. A José Enrique, por afianzar mi amor por las matemáticas ya existente desde mis primeros estudios escolares.

A mis amigos, los nuevos y los viejos.

A todos, gracias.

“La constancia no está en empezar sino en perseverar”.

Leonardo da Vinci

RESUMEN

En este trabajo se modela e implementa un sistema inteligente difuso para la evaluación del riesgo tecnológico en un área dada. La necesidad de la modelación borrosa viene dada por la presencia de vaguedad, imprecisión o subjetividad en los parámetros que determinan el índice de riesgo tecnológico los cuales son: Coeficiente de Agravamiento del Impacto, Vulnerabilidad Física y Frecuencia.

Usando la herramienta Xfuzzy 3.0, se definieron los términos lingüísticos que caracterizan las distintas variables para lo cual se trabajan funciones de pertenencia de tipo sigmas y campanas.

Se desarrollan seis sistemas de inferencia borrosos del tipo Mamdani para determinar los parámetros que determinan el índice de riesgo tecnológico con los correspondientes ajustes en el proceso de defuzzificación. Posteriormente se implementó una interfaz gráfica en Java para una mayor comodidad del usuario, realizándose las validaciones correspondientes.

Este sistema constituye un módulo del SEVAT (Sistema para la evaluación de la vulnerabilidad territorial ante accidentes tecnológicos graves) el cual ya ha sido incorporado para las pruebas de campo correspondiente. La adición de este módulo facilitará la evaluación del riesgo tecnológico y la toma de decisiones para mitigarlo, procedimiento anteriormente tratado manualmente y sujetos a los niveles de imprecisiones y subjetividad que implicaba el manejo de dichos factores.

ABSTRACT

In this work its modeled and its implements an intelligent fuzzy system for the evaluation of the technological risk in a given area. The necessity to use a fuzzy model comes given by the presence of uncertainty, imprecision or subjectivity in the parameters that determine the index of technological risk, which are: *Impact Coefficient Aggravation*, *Physical Vulnerability* and *Frequency*.

With the tool Xfuzzy 3.0, it was defined the linguistic terms that characterize the different variables using membership functions of type sigma and bells.

It was developed six Inference Fuzzy Systems of type Mamdani to determine the parameters that determine the index of technological risk with the corresponding adjustments in the defuzzification process. Subsequently, a graphic interface was implemented in Java for the user's bigger comfort, being carried out the corresponding validations.

This system constitutes a module of SEVAT (System for the evaluation of the territorial vulnerability before technological serious accidents) which has already been incorporate for the tests of corresponding field. The addition of this module will facilitate the evaluation of the technological risk and the decisions taking to mitigate it, procedure previously tried manually and subject to the imprecisions levels and subjectivity that it implied the handling of these factors.

TABLA DE CONTENIDO

DECLARACIÓN JURADA	ii
DEDICATORIA	iii
AGRADECIMIENTOS	iv
RESUMEN.....	vi
ABSTRACT.....	vii
TABLA DE CONTENIDO.....	viii
INTRODUCCIÓN	1
Problema de investigación	2
Objetivo general.....	2
Objetivos Específicos.....	2
Preguntas de investigación.....	2
Hipótesis de investigación.....	3
CAPÍTULO 1: APLICABILIDAD DE LA INTELIGENCIA ARTIFICIAL MEDIANTE LOS CONJUNTOS BORROSOS Y LA LÓGICA DIFUSA EN LA EVALUACIÓN DEL RIESGO TECNOLÓGICO	5
1.1 Definiciones básicas.....	5
1.2 Sistema de inferencia borroso	9
1.3 Ejemplos de solución de problemas utilizando la teoría de los conjuntos borrosos.	11
1.4 Tecnología utilizada para la modelación borrosa: Xfuzzy 3.0.	14
1.5 Conclusiones parciales	15
CAPÍTULO 2: MODELACIÓN BORROSA DEL ÍNDICE DE RIESGO TECNOLÓGICO E IMPLEMENTACIÓN COMPUTACIONAL DEL SISTEMA	16
2.1 Indicadores para la evaluación de riesgos tecnológicos	16
2.1.1 Coeficiente de agravamiento del impacto.	17
Falta de homeostasis.	18
Falta de resiliencia.....	18
2.1.2 Vulnerabilidad física.	19
2.1.3 Frecuencia de evento.....	20
2.2. Definición de las variables lingüísticas y términos lingüísticos.	20
Variable CAI:	20

Variable Vulnerabilidad:.....	27
Variable Frecuencia:	28
2.3 Modelación de los distintos SIB que constituyen el IRT	30
2.3.1 SIB AgravamientoImpacto.....	30
2.3.2 SIB Vulnerabilidad	32
2.3.3 SIB Frecuencia.....	34
2.4 Implementación de la interfaz en Java. SIDCIRT como módulo del software SEVAT	34
2.4.1 Creación del proyecto en Java e integración del XFuzzy	35
2.4.2 Diagrama de clases.....	36
2.4.3 Diagrama de componentes	37
2.4.4 Diagrama de actividades	38
2.5 Conclusiones parciales	40
CAPÍTULO 3: SISTEMA INTELIGENTE DIFUSO PARA EL CÁLCULO DEL RIESGO TECNOLÓGICO (SIDCIRT)	42
3.1 Requerimientos de la aplicación	42
3.2 Validación de SIDCIRT	42
3.3 Integración con SEVAT	43
3.4 Funcionamiento general de SIDCIRT	43
3.5 Conclusiones parciales	50
CONCLUSIONES	51
RECOMENDACIONES	52
REFERENCIAS BIBLIOGRÁFICAS	53
ANEXO I: Funciones de pertenencia utilizadas y métodos de defuzzificación. Definiciones del XFuzzy 3.0.	54
ANEXO II: Modelación borrosa en Xfuzzy 3.0.	57
ANEXO III: Diagramas de clases de SIDCIRT	67
ANEXO IV: Casos de estudio.....	73

INTRODUCCIÓN

Con el creciente desarrollo de las ciencias computacionales y dentro de ellas la inteligencia artificial, la aplicabilidad de esta se ha extendido a diversas esferas de la vida del hombre, las actividades socioeconómicas y a integrarse con otras ciencias. Existen disímiles ejemplos de su uso en la medicina, la industria, las actividades militares, la robótica, la informática y las telecomunicaciones; ya sea para diagnosticar, realizar automáticamente actividades que serían peligrosas, monótonas o de mucha precisión como para ser realizadas por el hombre, así también como manejar un avión no tripulado durante tareas de alto riesgo, el procesamiento del lenguaje natural o la minería de datos. Otro uso importante es la gestión medioambiental, ya sea mediante los predictores de desastres o sistemas evaluadores del riesgo de amenazas de origen natural como los fenómenos atmosféricos o los provocados por el hombre como los accidentes tecnológicos.

En los últimos años, desde la perspectiva de los desastres tecnológicos, el riesgo se ha intentado evaluar, para efectos de la gestión, en términos de las posibles consecuencias económicas, sociales y ambientales que pueden ocurrir en un lugar y en un tiempo determinado. La falta de una concepción holística del riesgo que permita degradarlo en sus componentes de diferente índole, parece haber contribuido en buena parte a la falta de efectividad de su gestión, por lo cual es necesario un mecanismo que permita valorarlo de manera integral. (Cardona, 2001)

Cualquier institución que se muestre vulnerable ante accidentes tecnológicos, sigue una política de gestión de riesgos que no es más que un enfoque estructurado para manejar la incertidumbre relativa a una amenaza, a través de una secuencia de actividades humanas que incluyen evaluación del mismo, estrategias de desarrollo para manejarlo y mitigación utilizando recursos gerenciales. (Arboleda, 2008)

La evaluación del riesgo es probablemente el paso más importante en un proceso de gestión de riesgos, y también el paso más difícil y con mayor posibilidad de cometer errores debido a la gran cantidad de variables que se utilizan para el proceso de evaluación, las cuales pueden ser continuas o discretas, sin embargo los especialistas se

refieren a ellas utilizando conceptos vagos e imprecisos tales como por ejemplo la *Densidad de Población* que puede ser *Alta*, *Media* o *Baja*.

Todo lo anterior, presupone el manejo adecuado de la incertidumbre que emana dichos conceptos, sugiriéndose por varios autores como una vía de solución la modelación borrosa.

Problema de investigación

Modelar la imprecisión y vaguedad en la evaluación de riesgos tecnológicos teniendo en cuenta la variedad de factores y la subjetividad presente en ellos.

Objetivo general

Diseñar un sistema inteligente difuso para la evaluación de riesgos tecnológicos en un área determinada, utilizando la herramienta Xfuzzy3.0 y con una interfaz en Java.

Objetivos Específicos

- Definir los factores que caracterizan el nivel de agravamiento, la vulnerabilidad física del área en estudio y la frecuencia del evento.
- Realizar la modelación borrosa de las diferentes variables que caracterizan a dichos factores utilizando la herramienta Xfuzzy3.0.
- Implementar una interfaz en Java para realizar el cálculo del índice de riesgo utilizando el sistema difuso modelado previamente.

Preguntas de investigación

- ¿Cuáles son los factores que caracterizan el nivel de agravamiento y la vulnerabilidad física del área en estudio?
- ¿Qué bondades ofrece Xfuzzy en cuanto a la modelación borrosa?
- ¿Qué aspectos se deben tener en cuenta a la hora de definir las reglas de inferencia borrosas?
- ¿Cómo integrar un SIB modelado con el Xfuzzy a un proyecto en Java?

Hipótesis de investigación

La modelación difusa para la evaluación del riesgo tecnológico, a partir de la valoración holística de sus distintos indicadores, contribuirá favorablemente a la toma de decisiones y a manejar adecuadamente la imprecisión y vaguedad de los mismos; partiendo del conocimiento y experiencia de los expertos en esta área.

A partir de esta hipótesis se proponen las siguientes tareas de investigación:

- Estudio profundo de la teoría de la lógica borrosa, así como las técnicas de construcción de sistemas de inferencia borrosos.
- Análisis de los factores que caracterizan el nivel de agravamiento del impacto del riesgo, la vulnerabilidad física del área expuesta y la frecuencia del evento.
- Estudio de las facilidades del software Xfuzzy para la modelación borrosa.
- Modelación de los sistemas de inferencia borrosos constituidos por estos factores mediante el Xfuzzy y validación de los mismos
- Implementación de una interfaz en Java con los sistemas difusos modelados previamente para la evaluación del riesgo.
- Evaluación del sistema con los expertos.

Mediante la lógica difusa podemos analizar los factores de riesgo tecnológicos desde una perspectiva holística, además de manejar adecuadamente la incertidumbre y la imprecisión de los conceptos. Un enfoque de este tipo podría contribuir a mejorar la efectividad de la gestión y a identificar y priorizar medidas factibles y eficientes de prevención-mitigación para la reducción del riesgo.

Para el desarrollo de la investigación se cuenta con gran cantidad de información científica alrededor del tema. Además, se utilizarán las herramientas *NetBeans7.2.1* para la interfaz visual en Java de la aplicación y el entorno de desarrollo para sistemas de inferencias borrosos *XFuzzy3.0*, que son software libre y multiplataforma, permitiendo la ejecución de la misma en todos los sistemas operativos que soporten un entorno de ejecución de java (JRE por sus siglas en inglés).

Finalmente, en el grupo de investigación de Inteligencia Artificial se cuenta con los medios técnicos y con el software necesario para poder realizar todo el trabajo de implementación.

La tesis está estructurada en tres capítulos. El primero se refiere a los conceptos y definiciones sobre el riesgo tecnológico, así como el marco teórico de las técnicas de inteligencia artificial empleadas en la elaboración del sistema inteligente difuso. En el segundo se explica la modelación borrosa de los diferentes indicadores del índice de riesgo tecnológico. Para finalizar, el tercer capítulo aborda sobre la implementación, documentación y prueba del sistema.

CAPÍTULO 1: APLICABILIDAD DE LA INTELIGENCIA ARTIFICIAL MEDIANTE LOS CONJUNTOS BORROSOS Y LA LÓGICA DIFUSA EN LA EVALUACIÓN DEL RIESGO TECNOLÓGICO

Este capítulo aborda los conceptos básicos relacionados con la lógica difusa necesarios para la modelación del sistema. Se exponen las características del software Xfuzzy3.0, donde se realizó la modelación borrosa y se muestran ejemplos de soluciones a problemas mediante la lógica borrosa.

1.1 Definiciones básicas

En los años setentas se reconoció que para resolver los problemas orientados a aplicaciones, los métodos de solución de problemas generales eran insuficientes, ya que era necesario conocimiento específico sobre el problema, limitado a los dominios de aplicación de interés, en lugar de conocimiento general aplicable a muchos dominios. Este reconocimiento condujo al desarrollo de Sistemas Basados en el Conocimiento (SBC). (Lio, 1998)

El conocimiento representado en los SBC es el de los expertos en el dominio. Una parte de un conocimiento experto consiste de relaciones de causa y efecto. Estas relaciones o reglas se originan a partir de la experiencia pasada y son llamadas heurísticas. Un experto es una persona que posee habilidades que le permiten sacar conclusiones de experiencias pasadas y rápidamente focalizar sobre el centro de un problema dado. Las mayores posibilidades del éxito de un experto en la solución de problemas se deben a que ha adquirido un conjunto de relaciones de causa y efecto poderosas basadas en la experiencia. Un experto es capaz de utilizar este conocimiento básico para reconocer rápidamente rasgos sobresalientes del problema, clasificar este de acuerdo a estas características y buscar una solución. (Lio, 1998)

Por lo tanto un SBC se define como un sistema computarizado que usa conocimiento sobre un dominio para arribar a una solución de un problema de ese dominio. Esta

solución es esencialmente la misma que la obtenida por una persona experimentada en el dominio del problema cuando se enfrenta al mismo problema. (Lio, 1998)

Los SBC, según (Lio, 1998), se distinguen de los programas basados en búsqueda general principalmente por:

1. La separación del conocimiento de cómo este es usado (distinción entre conocimiento y estrategia de control).
2. El uso de conocimiento muy específico del dominio.
3. Naturaleza heurística, en lugar de algorítmica, del conocimiento empleado.

Otros rasgos que lo distinguen de la programación convencional son que no requieren analizar completitud y pueden dar diversas soluciones. Además permiten el tratamiento de los rasgos continuos como discretos o difusos, donde juegan un papel fundamental los Sistemas de Inferencia Borroso como un caso particular de SBC para tratar problemas en presencia de vaguedad o incertidumbre.

Durante el proceso de evaluación del riesgo tecnológico nos encontramos que este depende de variables que son de diferentes dominios y generalmente se hablan de ellas manejando conceptos vagos como por ejemplo en el **Nivel de desarrollo humano** si es *Alto* o *Bajo* o si la **Protección** es *Buena* o *Mala*.

Para tratar problemas en presencia de vaguedad o incertidumbre, los conjuntos borrosos y los sistemas contruidos a partir de ellos constituyen útiles herramientas.

Definición 1: Conjuntos borrosos y funciones de pertenencia. (Bello, 1998)

Si X es una colección de objetos denotados genéricamente por x , entonces un conjunto borroso A en X se define como un conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \text{ (Turban, 1992)}$$

Donde $\phi_A(x)$ es llamada la función de pertenencia (MF) para el conjunto A. La MF asigna a cada elemento de X un grado de pertenencia en el intervalo [0,1]. A X se llama universo de discurso y puede ser un espacio discreto o continuo.

Definición 2: Variables lingüísticas. (Bello, 1998)

Una variable lingüística se caracteriza por un quintuplo $(x, T(x), X, G, M)$ en el cual x es el nombre de la variable, $T(x)$ es el conjunto de términos, o sea, el conjunto de sus valores o términos lingüísticos, X es el universo de discurso, G es la regla sintáctica la cual genera los términos en $T(x)$, y M es una regla semántica la cual asocia a cada valor lingüístico A su significado $M(A)$, donde $M(A)$ denota un conjunto borroso en X.

Definición 3: Razonamiento borroso. (Bello, 1998)

El razonamiento borroso (RB), también conocido como razonamiento aproximado, es un procedimiento de inferencia que deriva conclusiones a partir de un conjunto de reglas borrosas y hechos conocidos.

El RB o modus ponens generalizado (GMP) se describe como:

Premisa 1 (hecho): x is A',

Premisa 2 (regla): If x es A then y es B,

Conclusión: y es B'.

Donde A' está cercano a A y B' a B; A, B, A' y B' son conjuntos borrosos de universos apropiados.

Definición 4: (Bello, 1998)

Sean A, A' y B conjuntos borrosos de X, X e Y respectivamente. Asumimos que la implicación borrosa $A \rightarrow B$ se expresa como una relación borrosa R sobre $X \times Y$.

Entonces el conjunto borroso B' inducido por " x es A' " y la regla borrosa "If x es A then y es B " se define por:

$$\varphi_{B'}(y) = \max \min [\varphi_{A'}(x), \varphi_R(x,y)] = \forall x [\varphi_{A'}(x) \text{ and } \varphi_R(x,y)]$$

En el caso más simple, cuando se tiene una sola regla como la anterior, es decir, con un antecedente simple, la expresión anterior se puede reducir a:

$$\varphi_{B'}(y) = [\forall x (\varphi_{A'}(x) \text{ and } \varphi_A(x))] \text{ and } \varphi_B(y)$$

$$= w \text{ and } \varphi_B(y)$$

es decir, primero se encuentra el valor w como el máximo de $\varphi_{A'}(x)$ y $\varphi_A(x)$, entonces la MF de M' es igual a la MF de B acotada por w . Intuitivamente, w representa una medida del grado de creencia sobre el antecedente de la regla, es decir, denota el grado de compatibilidad entre A y A' .

Otro caso es cuando se tiene una regla con múltiples antecedentes; por ejemplo, una regla borrosa con dos antecedentes es "If x es A and y es B then z es C ".

El conjunto borroso C' queda definido por:

$$\varphi_{C'}(z) = (\forall x [\varphi_{A'}(x) \text{ and } \varphi_A(x)]) \text{ and } (\forall y [\varphi_{B'}(y) \text{ and } \varphi_B(y)]) \text{ and } \varphi_C(z)$$

$$= (w_1 \text{ and } w_2) \text{ and } \varphi_C(z)$$

donde w_1 y w_2 son los máximos de las MF de $A \cap A'$ y $B \cap B'$ respectivamente, y $(w_1 \text{ and } w_2)$ se denomina fortaleza de la activación de la regla o grado de satisfacción de la regla borrosa.

Cuando los antecedentes de la regla se conectan por OR en lugar de AND, entonces se usa el máximo en lugar del mínimo para calcular los w_i .

1.2 Sistema de inferencia borroso

Un sistema de inferencia borroso (SIB) es un sistema computacional basado en los conceptos de la teoría de conjuntos borrosos, reglas If-Then borrosas y razonamiento borroso. Se conocen por diversos nombres como Sistemas basados en reglas borrosos, Sistemas expertos borrosos, etc. (Morell and Bello, 2007)

La estructura básica de un SIB consiste de tres componentes: una base de reglas borrosas, una base de datos (la cual contiene las funciones de pertenencia usadas) y un mecanismo de razonamiento el cual ejecuta el procedimiento de inferencia. (Morell and Bello, 2007)

Las entradas y salidas pueden ser valores duros o borrosos. Cuando la salida es borrosa y se necesita el valor duro, se emplea un método (*defuzzificación*) que determina el valor duro que mejor representa un conjunto borroso.

Métodos de *defuzzificación*:

Es el proceso de encontrar el valor del universo que mejor “represente” al conjunto borroso. Para ello se traza una recta perpendicular al eje de las abscisas (donde se representa el universo del conjunto borroso), el punto donde se interceptan esta recta y ese eje es el valor que se asume. Los diferentes métodos determinan el lugar por donde se traza esa recta. Se pueden emplear cinco métodos, (Bello, 1998):

- Centroide del área: encuentra el punto “balance” de la región borrosa solución calculando la media pesada de esa región borrosa. Dada una región borrosa A, la expresión de cálculo es:

$$z = \frac{\sum_{i=0}^n d_i * \varphi_A(d_i)}{\sum_{i=0}^n \varphi_A(d_i)} \text{ donde } d \text{ es el } i\text{-ésimo valor del universo y } \varphi(d) \text{ es el valor de}$$

pertenencia para ese punto. Este método es uno de los más usados por varias razones: los valores tienden a moverse suavemente alrededor de la superficie borrosa, es decir, cambios en la topología del conjunto borroso provocan cambios suaves del valor defuzzificado; por otra parte, es relativamente fácil de calcular.

- Bisector del área.
- Medio de máximos: es el promedio de los valores del universo de discurso donde se alcanza el máximo grado de membresía.
- Mínimo de los de máximos: es el valor menor del universo de discurso con el cual se alcanza el máximo grado de membresía.
- Máximo de los máximos: es el valor mayor del universo de discurso con el cual se alcanza el máximo grado de membresía.

Existen otros tipos de métodos de defuzzificación pero estos son los más conocidos y por lo tanto los candidatos para nuestros SIB, probando con cada uno de ellos hasta obtener el que arroje mejores resultados.

Tipos de Sistemas de Inferencia Borrosos:

Existen tres tipos de SIB, que se diferencian en la forma del consecuente de sus reglas borrosas y por eso varían los procedimientos de agregación y defuzzificación, (Bello, 1998).

Es importante resaltar una característica de los SIB que lo distinguen de los Sistemas Basados en Reglas: el resultado de la inferencia se obtiene de aplicar numerosas reglas (usualmente todas las de la base) y conciliar las inferencias parciales de estas, (Bello, 1998).

La forma en que se concilian o integran todas estas inferencias parciales para producir un valor se denomina procedimiento de agregación. El producto de la agregación es el que posiblemente sea necesario defuzzificar, (Bello, 1998).

- *Modelo borroso de Mamdani:*

En este modelo el antecedente y el consecuente de las reglas son conjuntos borrosos; la regla puede tener múltiples antecedentes, por esta razón cumple con los requisitos necesarios para modelar nuestro problema y por lo tanto será el utilizado.

El método de inferencia de este modelo se basa en el empleo de un esquema de composición para producir el conjunto borroso final. Este esquema de composición indica como acotar el conjunto borroso consecuente de cada regla a partir de los valores de sus antecedentes y luego como integrar (agregar) todos los conjuntos borrosos resultantes en uno solo. Se reconocen dos esquemas de combinación el max-min y el max-producto. En ambos se utiliza el operador max (unión) para realizar la agregación de los conjuntos borrosos. Difieren en la forma de hallar el acotamiento del conjunto borroso que resulta de cada regla, en el primero se usa el operador mínimo y en el segundo el operador producto, (Bello, 1998).

- *Modelo de Sugeno.*
- *Modelo borroso de Tsukamoto.*

Modelación borrosa:

La modelación borrosa es el proceso de construir un SIB. La modelación borrosa se ejecuta en dos etapas. En la primera se identifica la estructura superficial del sistema y en la segunda la estructura profunda, (Morell and Bello, 2007).

Estas etapas se verán de forma concreta en el siguiente capítulo, donde se realiza la modelación de los distintos SIB que caracterizan al IRT.

1.3 Ejemplos de solución de problemas utilizando la teoría de los conjuntos borrosos.

La aplicabilidad de la lógica borrosa es muy amplia, según (Cordón), se han realizado aplicaciones para:

- La estimación de los costes de mantenimiento de las líneas eléctricas de baja y media tensión, donde se utilizan cuatro variables de entrada: longitudes de las calles, área total, área ocupada por edificios y energía suministrada y una variable de salida: costo de mantenimiento de la línea de baja o media tensión.
- La valoración de la calidad del arroz cocido, donde se utilizan cinco variables de entrada: gusto, apariencia, sabor, pegajosidad y dureza.

- La predicción del riesgo de contraer enfermedades cardiovasculares donde se utilizan como variables de entrada: nivel de colesterol total en plasma, nivel de LDL-colesterol en plasma, nivel de HDL-colesterol en plasma, nivel de triglicéridos en plasma, edad del individuo y hábitos del individuo, como variable de salida se tiene la predicción del riesgo de contraer una enfermedad cardiovascular.
- Problema del sonar: donde un barco, situado en la superficie, lanza una señal de sonar contra el fondo marino y recoge el eco de la misma, se trata de clasificar los objetos detectados en dos posibles clases: cilindros metálicos y rocas con esa forma, se recogen 60 características de la señal de sonar (60 variables de entrada), la base de datos contiene 208 ejemplos del problema.

Según la conferencia de la *IEEE Industrial Applications Society* realizada en Panamá en 2009 sobre las aplicaciones de la lógica difusa, (Society, 2009), estas se encuentran presentes en distintos campos como:

- Ahorro de Energía: Control de Lavarropas. Control adaptativo de Sistemas de Calefacción.
- Industria Automotriz: Sistema de Frenos Antibloqueo. Control de Motores de Combustión Interna. Control de Crucero. Control de Aire Acondicionado para Automóviles.
- Automación Industrial: Control de Decantador. Control de Producción de Plásticos. Control de Reciclado de Agua. Control de Horno de Cocción de bizcochos.
- Análisis de Datos: Aplicaciones Médicas. Aplicaciones Sensoriales.
- Otras Aplicaciones: Cargador de Baterías. Estabilización de Imágenes Digitales en Cámaras. Control de Anestesia.

Otro ejemplo de aplicación es el trabajo de (Mata-García et al., 2006). El proyecto consiste en un sistema de riego inteligente utilizando lógica borrosa que permite simular por software la evolución de un entorno de regadío simple en el tiempo, permitiendo modificar ciertas condiciones ambientales y posibilitando la elección de realizar

inferencias para el cálculo del tiempo de riego. Cuenta con un entorno de simulación en Java para visualizar gráficamente el comportamiento de la aplicación frente a la variación de los distintos parámetros de entrada, ya sea de manera instantánea, diaria o mensual. También muestra un estudio comparativo de la respuesta de este sistema frente a otros sistemas habituales utilizados en el regadío. Las variables que utiliza como entrada son: radiación solar (incidencia de rayos solares), nubosidad (hace variar la incidencia de los rayos solares), humedad relativa (% de agua en la tierra), superficie foliar de la planta (para conocer el crecimiento de la planta), viento (varía la evaporación y transpiración de la planta), precipitación (para controlar el riego), permeabilidad del suelo (permite control del encharcamiento), calidad del agua (según condiciones químicas) y presión y caudal de riego. Para la modelación borrosa se utilizó el software Xfuzzy 3.0, que es precisamente el mismo entorno que utilizaremos para nuestro trabajo.

En nuestro país también podemos encontrar aplicaciones de la lógica borrosa como es el modelo para la ayuda a la toma de decisiones basado en técnicas de *softcomputing* y su uso en el diagnóstico médico, (Piñero et al., 2006), desarrollado por investigadores de la Universidad de Ciencias Informáticas (UCI) y especialistas médicos. Este trabajo presenta un nuevo modelo que permite aprender y generalizar en forma de reglas borrosas el conocimiento implícito en bases de casos y que posibilita además la clasificación de nuevos casos tomando como base la explotación de un sistema de inferencia borroso del tipo *Sugeno* grado cero. Como parte del modelo desarrollado se presenta el algoritmo *MLRul* para la generación de reglas borrosas. Se desarrollan además algoritmos para la optimización de bases de reglas borrosas tomando como base la aplicación de heurísticas y algoritmos evolutivos. Se muestra la aplicación del modelo desarrollado sobre bases de datos internacionales utilizando el sistema *MLClassif* y se comparan los resultados con otros sistemas y modelos para la clasificación.

1.4 Tecnología utilizada para la modelación borrosa: Xfuzzy 3.0.

Xfuzzy 3.0 es un entorno de desarrollo para sistemas de inferencia basados en lógica difusa. Está formado por varias herramientas, incluyendo herramientas gráficas, que cubren las diferentes etapas del proceso de diseño de sistemas difusos, desde su descripción inicial hasta la implementación final. Sus principales características son la capacidad para desarrollar sistemas complejos y la flexibilidad para permitir al usuario extender el conjunto de funciones disponibles. El nexo entre todas las herramientas es el uso del lenguaje de especificación *XFL3*, el cual es un lenguaje flexible y potente que permite expresar relaciones muy complejas entre variables difusas por medio de bases de reglas jerárquicas y conectivas, modificadores lingüísticos, funciones de pertenencia y métodos de defuzificación definidos por el usuario. El entorno ha sido completamente programado en Java, de forma que puede ser ejecutado sobre cualquier plataforma que tenga instalado el JRE (*Java Runtime Environment*), (IMSE-CNM, 2003b).

El Xfuzzy presenta algunos problemas en su editor gráfico, en algunas funcionalidades que no se desempeñan correctamente, pero ante esta situación se puede siempre editar el archivo *XFL* y hacer a mano lo deseado, lo cual funciona sin problema.

Una de las facilidades del software Xfuzzy y por lo tanto uno de los motivos por el cual es elegido para la implementación de nuestro sistema es que genera por cada SIB un archivo *.xfl*, como se muestra en la figura 1.1 y brinda la facilidad de trasladarlo a código Java, C o C++, lo cual permite realizar posteriormente una interfaz de usuario:

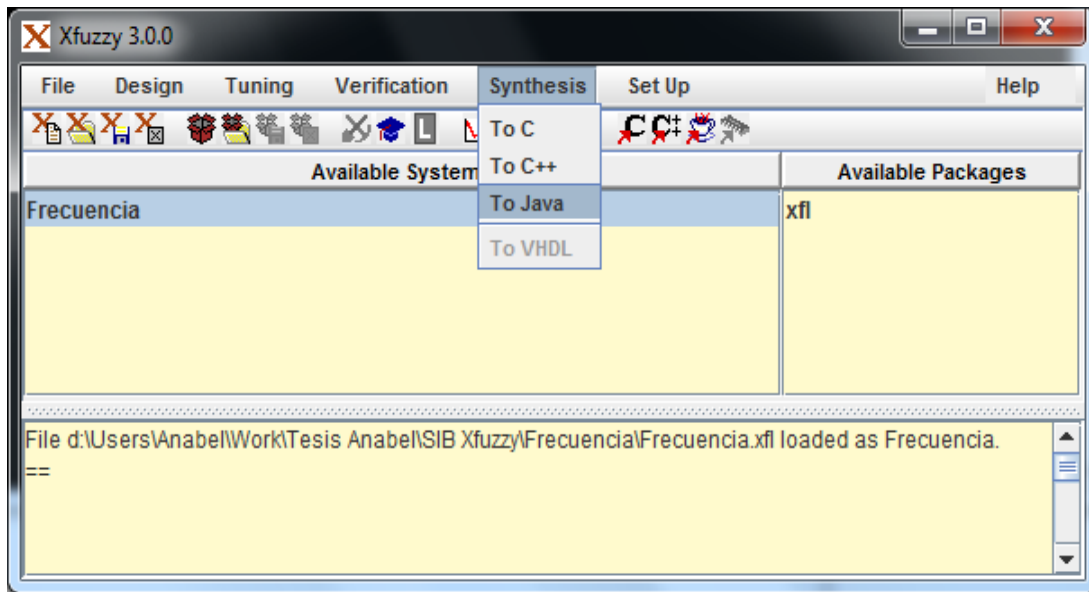


Figura 1.1

1.5 Conclusiones parciales

- La teoría de los conjuntos borrosos y la lógica difusa son herramientas de gran importancia para modelar el sistema y realizar una correcta evaluación del riesgo tecnológico.
- El software Xfuzzy3.0 constituye una herramienta útil para la definición de SIB de avanzada complejidad y para el proceso de ajuste de parámetros.

CAPÍTULO 2: MODELACIÓN BORROSA DEL ÍNDICE DE RIESGO TECNOLÓGICO E IMPLEMENTACIÓN COMPUTACIONAL DEL SISTEMA

El presente capítulo describe cómo se implementaron los distintos SIB necesarios para el cálculo del IRT utilizando la herramienta XFuzzy3.0, así como su integración con Java para la implementación de la interfaz del sistema.

Para la construcción de los SIB mencionados se siguieron los pasos propuestos por (Bello, 1998):

Primera etapa:

- Seleccionar las variables de entrada y salida relevantes.
- Seleccionar un tipo específico de SIB.
- Determinar la cantidad de términos lingüísticos asociados con cada variable de entrada o salida.
- Definir el conjunto de reglas borrosas.

Segunda etapa:

- Seleccionar una familia apropiada de MF parametrizadas.
- Entrevistar a los expertos humanos para determinar los parámetros de las MF usadas.
- Refinar los parámetros de las MF usando técnicas de regresión y optimización (si hay un conjunto de datos de entrada-salida).

2.1 Indicadores para la evaluación de riesgos tecnológicos

El índice de riesgo total (**IRT**) de un área dada se puede expresar, como la sumatoria de la frecuencia de ocurrencia del evento (**Fe**) por la vulnerabilidad física del área en estudio (**VFe**) ante el evento (**e**), para cada evento posible (**e**), en nuestro caso se estudian cuatro: incendio, explosión, derrame y escape. Finalmente toda la sumatoria se

afecta por un coeficiente de agravamiento del impacto (**CAI**), el cual caracteriza al área en estudio por su capacidad intrínseca de enfrentarse y reponerse ante un evento catastrófico y cuyos valores se ponderan entre 0 y 1. (Valdivia et al., 2010)

$$IRT = (1 + CAI) \sum_{e=1}^n (F_e * VF_e) \quad (I)$$

Siguiendo la fórmula anterior, el **IRT** presenta valores entre 0 y 8, sin embargo se utiliza la normalización de este resultado para que se pondere entre 0 y 1 también, ya que los especialistas utilizan esta terminología y realizan la clasificación del riesgo en este rango y se deciden las acciones a tomar, según (Valdivia et al., 2010) este se cataloga como:

- De 0 a 0.025: **Trivial**. No se requiere acción específica.
- De 0.025 a 0.10: **Tolerable**. No se necesita mejorar la acción preventiva, sin embargo se deben considerar soluciones más rentables o mejoras que no supongan una carga económica importante. Se requieren comprobaciones periódicas para asegurar que se mantiene la eficacia de las medidas de control.
- De 0.10 a 0.29: **Moderado**: Se deben hacer esfuerzos para reducir el riesgo, determinando las inversiones precisas, las medidas deben implementarse en un período determinado.
- De 0.29 a 0.57: **Importante**. No debe comenzarse el trabajo hasta que se haya reducido el riesgo. Puede que se precisen recursos considerables para controlarlo.
- De 0.57 a 1: **Intolerable**. No debe comenzar ni continuar el trabajo hasta que se reduzca el riesgo, si esto no es posible, incluso con recursos ilimitados, debe prohibirse el trabajo.

2.1.1 Coeficiente de agravamiento del impacto.

El coeficiente de agravamiento del impacto (CAI) depende de dos factores: **la falta de homeostasis** y **la falta de resiliencia**. Este coeficiente es independiente del tipo de amenaza y se determina para cada una de las áreas expuestas al riesgo, (Valdivia et al., 2010).

Las áreas expuestas pueden estar compuestas por viviendas particulares, escuelas, centros laborales, hospitales, círculos infantiles, hogares de ancianos, industrias, etc.

Se describirán ahora estos factores.

Falta de homeostasis.

La falta de homeostasis no es más que la falta de capacidad para no afectarse y depende de los siguientes parámetros, (Valdivia et al., 2010):

- **Densidad de población:** Es la cantidad de habitantes (trabajadores y población aledaña) por kilómetros cuadrados en el área del riesgo.
- **Área comunitaria:** Es un cuadrante de $1 \times 1 \text{ km}^2$ compuesta por viviendas particulares, escuelas, centros laborales, hospitales, círculos infantiles, hogares de ancianos, industrias, etc.
- **Preparación:** es la protección en el momento de la emergencia y depende de si se cuenta con un sistema de alerta temprana, comunidad entrenada para protegerse, comunicación, etc.
- **Protección:** es el resguardo del área expuesta mediante barreras naturales (montañas, ríos, bosques protectores, etc.) y/o barreras artificiales (muros, taludes, edificaciones, etc.).
- **Receptores ambientales vulnerables:** son los espacios naturales protegidos, especies protegidas y patrimonios histórico-artísticos que se encuentran expuestos.
- **Estado físico de las construcciones:** se analizan las condiciones en que se encuentran las edificaciones del área en estudio y su composición.

Falta de resiliencia.

La falta de resiliencia es la falta de capacidad para reponerse y depende de los siguientes factores, (Valdivia et al., 2010):

- **Servicios de salvamentos:** es la existencia de brigadas de lucha contra incendios (compuesta por los mismos trabajadores y vecinos del área), servicios de bomberos y ambulancias.

- **Operatividad en las emergencias:** Es la rapidez de respuesta de la brigada de lucha contra incendios, los bomberos y las ambulancias ante el evento y depende de la distancia y el tiempo en mitigar el evento por parte de estos servicios.
- **Servicios hospitalarios:** Depende de la eficiencia de los servicios de urgencias.
- **Recuperación básica:** es lograr las condiciones esenciales de subsistencia para todos y garantizar los servicios básicos: agua, alimentos, electricidad, comunicación, etc.
- **Nivel de desarrollo humano:** Representa el nivel de desarrollo de la población teniendo en cuenta los promedios de longevidad, alfabetización de las personas, su escolaridad y el ingreso en paridades de poder adquisitivo per cápita. A mayor desarrollo mayor es la capacidad para reducir el riesgo y enfrentar los desastres.
- **Reconstrucción:** consiste en recuperar iguales o mejores condiciones que las previas, o sea, rehacer el medio y las condiciones de subsistencia.

2.1.2 Vulnerabilidad física.

La vulnerabilidad física es el grado de pérdida de un elemento o grupo de elementos bajo riesgo, resultado de la probable ocurrencia de un suceso desastroso, expresada en una escala desde 0 (sin daño) a 1 (pérdida total). Esta depende de los siguientes factores, (Valdivia et al., 2010):

- **Gravedad del índice de fallecidos:** valora qué tan alarmante es el índice de personas fallecidas de todas las expuestas (trabajadores y población aledaña).
- **Gravedad del índice de lesionados:** valora qué tan alarmante es el índice de personas lesionadas de todas las expuestas (trabajadores y población aledaña).
- **Daños internos mayores:** indica el nivel de gravedad de los objetos constructivos internos con daños mayores ya sean instalaciones tecnológicas, de servicios e infraestructuras dentro del área industrial.
- **Daños externos mayores:** indica el nivel de gravedad de los objetos constructivos externos con daños mayores ya sean viviendas, infraestructuras sociales y áreas comunitarias aledañas.

- **Daños medioambientales:** indica el nivel de gravedad según el Índice Global de Consecuencias Medioambientales (IGCM).

2.1.3 Frecuencia de evento.

La frecuencia del evento depende del tiempo de demora de ocurrencia entre un evento y otro, (Valdivia et al., 2010).

2.2. Definición de las variables lingüísticas y términos lingüísticos.

Variable CAI:

- **outAI:** Es la variable principal de nuestro sistema ya que es la salida de este y se refiere al Agravamiento del Impacto. Su universo de discurso es $[0,1]$ y los términos lingüísticos: **Bien**, **Regular**, **Mal**

Las funciones de pertenencia se construyeron utilizando las funciones matemáticas sigmas y campanas debido a que son las que más se adecuan a los requerimientos del usuario. En la mayoría de las variables que utilizan los términos lingüísticos **Bien**, **Regular** o **Mal**, donde el universo de discurso es $[0,1]$ las funciones se construyen de la siguiente forma:

Bien se representa por una función sigma y los parámetros a y b de la misma se evalúan por 0.25 y -0.01; en el caso de **Regular** se representa por la función campana con los valores de a y b de 0.5 y 0.2 respectivamente y en el caso de **Mal** es una sigma cuyos parámetros se evalúan con los valores 0.75 y 0.01 respectivamente. La figura 2.1, 2.2 y 2.3 muestra en rojo el comportamiento de los términos lingüísticos **Bien**, **Regular**, **Mal**.

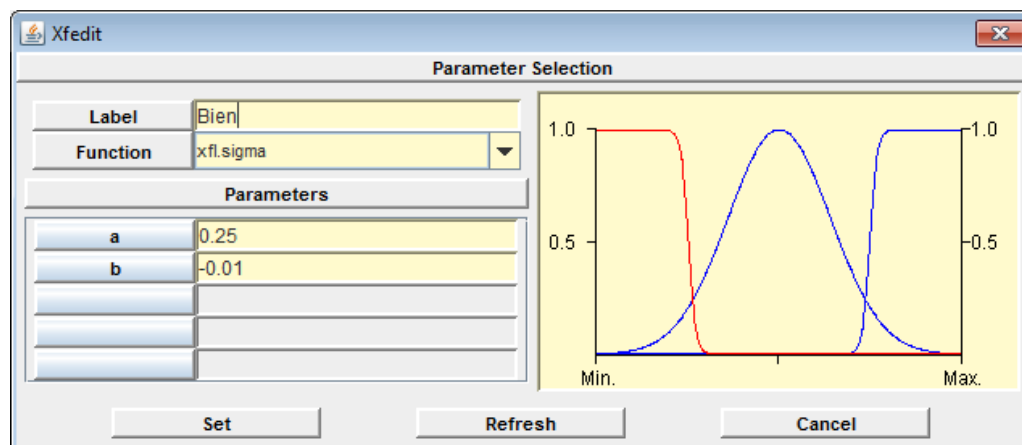


Figura 2.1. Funciones de pertenencia de los términos lingüísticos de la variable **outAI** (**Bien** en rojo)

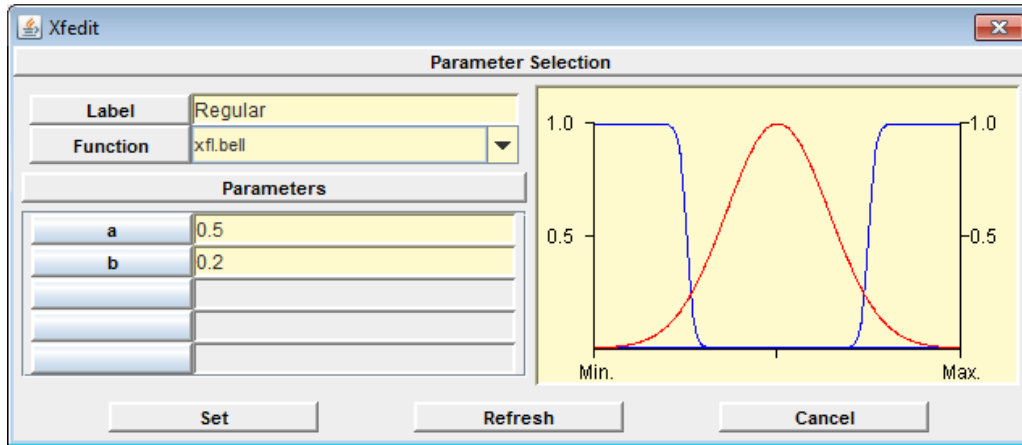


Figura 2.2. Funciones de pertenencia de los términos lingüísticos de la variable **outAI** (**Regular** en rojo)

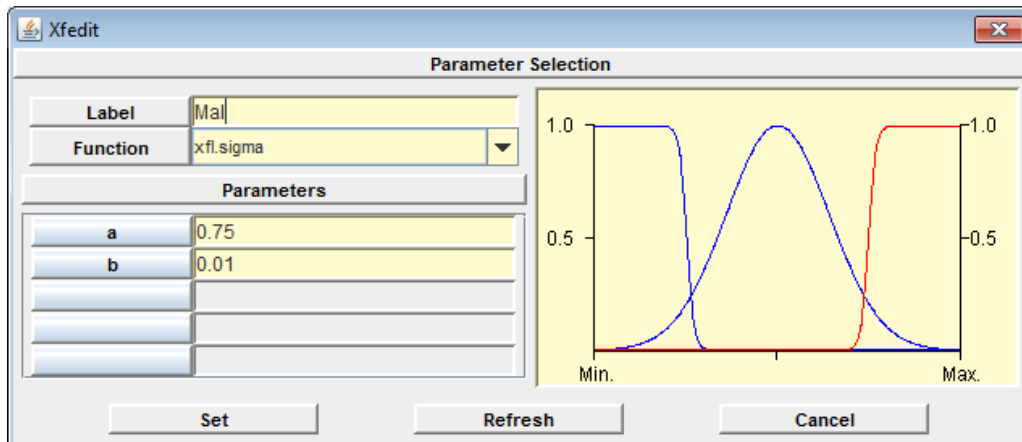


Figura 2.3. Funciones de pertenencia de los términos lingüísticos de la variable **outAI** (**Mal** en rojo)

La variable **outAI** se defuzzifica debido a que nos interesa obtener su valor duro, como método de defuzzificación se emplearon varios como el centroide del área, el máximo de los máximos y la media de los máximos, siendo este último el que mejores resultados arrojó y por lo tanto el elegido.

- **outFaltaHomeostasis**: la falta de homeostasis es una de las dos variables de las cuales depende el Agravamiento del Impacto, su universo de discurso es [0,1] y se modela con los términos lingüísticos: **Bien**, **Regular**, **Mal**, de forma análoga a la variable **outAI**.

Para su obtención se requieren conocer de las siguientes seis variables, las cuales están descritas por los términos *Bien, Regular o Mal* y su universo de discurso es $[0,1]$; excepto la **inDP** (Densidad de Población) cuyos términos lingüísticos son: *Alta, Media o Baja* y su universo de discurso $[0,500]$.

1. **outPrep:** Se refiere a la Preparación y esta a su vez depende de:

- ✓ **inSAT:** Se refiere al Sistema de Alerta Temprana.
- ✓ **inPME:** Se refiere a la Protección en el Momento de Emergencia.
- ✓ **inCEP:** Se refiere a la Comunidad Entrenada para Protegerse

Los términos lingüísticos asociados a estas 3 variables son: *Bien, Regular o Mal* y el universo de discurso de las mismas es de 0 a 1. Funciones de pertenencia similares a las de **outAI**.

2. **outProtec:** Se refiere a la Protección cuyos términos lingüísticos son *Bien, Regular o Mal* y el universo de discurso es de 0 a 1. Funciones de pertenencia similares a las de **outAI** y depende de:

- ✓ **inBN:** Se refiere a las Barreras Naturales.
- ✓ **inBA:** Se refiere a las Barreras Artificiales.

Los términos lingüísticos asociados a estas 2 variables son: *Eficientes, Medianamente Eficientes o Deficientes* y el universo de discurso de las mismas es de 0 a 5 (expresa tamaño de la barrera medida en metros). A continuación se muestran las funciones de pertenencia en la figura 2.4 donde *Eficientes* se representa por una función sigma y los parámetros a y b de la misma se evalúan por 3.0 y -0.01 ; en el caso de *Medianamente Eficientes* se representa por la función campana con los valores de a y b de 2.25 y 0.8 respectivamente y en el caso de *Deficientes* es una sigma cuyos parámetros se evalúan con los valores 1.25 y -0.01 respectivamente:

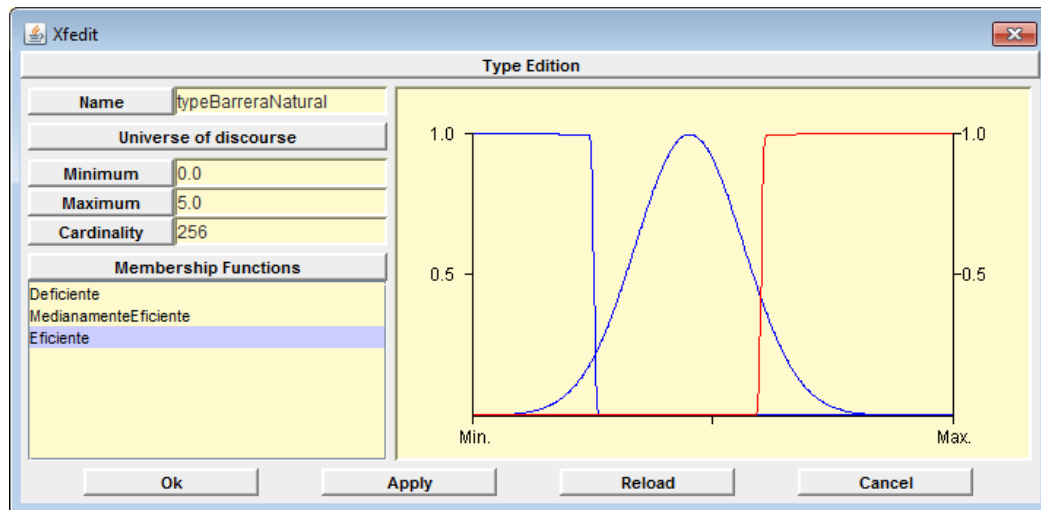


Figura 2.4

3. **outRAVul:** Se refiere a los Receptores Ambientales Vulnerables, los términos lingüísticos son *Bien*, *Regular* o *Mal* y el universo de discurso es de 0 a 1. Funciones de pertenencia similares a las de **outAI** y las variables de las cuales depende presentan las mismas características:
 - ✓ **inENP:** Se refiere a los Espacios Naturales Protegidos.
 - ✓ **inEP:** Se refiere a las Especies Protegidas.
 - ✓ **inPHA:** Se refiere al Patrimonio Histórico Artístico.
4. **inEFC:** Se refiere al Estado Físico de las Construcciones, los términos lingüísticos son *Bien*, *Regular* o *Mal* y el universo de discurso es de 0 a 1. Funciones de pertenencia similares a las de **outAI**.
5. **inDP:** Se refiere a la Densidad de Población y su universo de discurso es [0,500] (expresa habitantes por kilómetros cuadrados). A continuación se muestran las funciones de pertenencia en la figura 2.5 donde *Alta* se representa por una función sigma y los parámetros a y b de la misma se evalúan por 200.0 y 10.0; en el caso de *Media* se representa por la función campana con los valores de a y b de 150.0 y 50.0 respectivamente y en el caso de *Baja* es una sigma cuyos parámetros se evalúan con los valores 100.0 y -10.0 respectivamente:

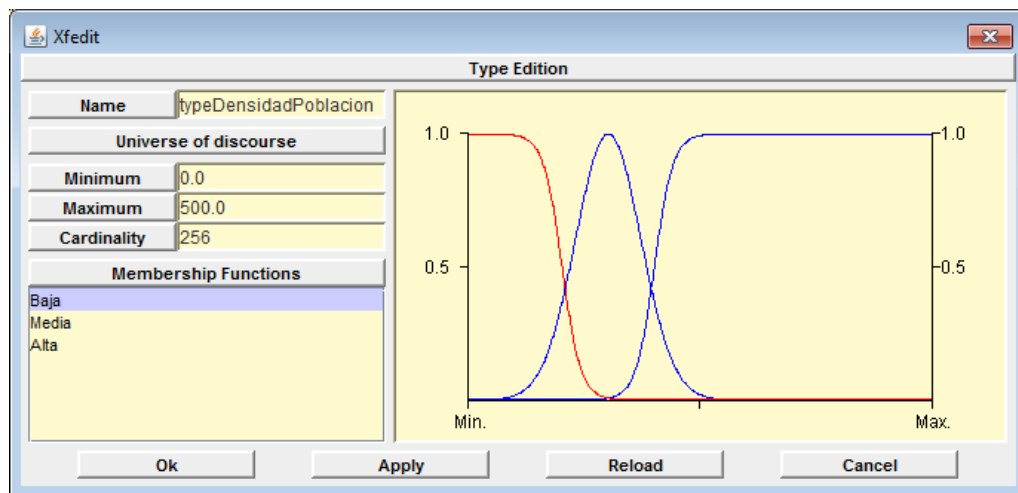


Figura 2.5

6. **inAC:** Se refiere al Área Comunitaria, los términos lingüísticos son *Bien*, *Regular* o *Mal* y el universo de discurso es de 0 a 1. Funciones de pertenencia similares a las de **outAI**.
- **outFaltaResiliencia:** Se refiere a la Falta de Resiliencia y es la otra variable de la cual depende el Agravamiento del Impacto, su universo de discurso es [0,1] y también sus términos son: *Bien*, *Regular* y *Mal*, las funciones de pertenencia son similares a las de outAI. Esta variable depende de otras seis variables y son:
 1. **inSS:** Se refiere a los Servicios de Salvamento, su universo de discurso es [0,1] y los términos son: *Bien*, *Regular* y *Mal*, las funciones de pertenencia son similares a las de outAI.
 2. **outOE:** Se refiere a la Operatividad de las Emergencias, su universo de discurso es [0,1] y los términos son: *Bien*, *Regular* y *Mal*, las funciones de pertenencia son similares a las de outAI. Esta a su vez depende de las siguientes variables las cuales utilizan para su descripción los siguientes términos lingüísticos: *Máxima Operatividad*, *Media Operatividad* o *Baja Operatividad*.
 - ✓ **inBLI:** Se refiere a las Brigadas de Luchas contra Incendios y su universo de discurso es [0,60] (se expresa en minutos). A continuación se muestran las funciones de pertenencia en la figura 2.6 donde *Máxima Operatividad* se representa por una función sigma y los parámetros a y b de la misma se evalúan por 9.0 y -1.0; en el caso de *Media Operatividad* se representa por la función

campana con los valores de a y b de 15.0 y 5.0 respectivamente y en el caso de **Baja Operatividad** es una sigma cuyos parámetros se evalúan con los valores 20.0 y 1.0 respectivamente:

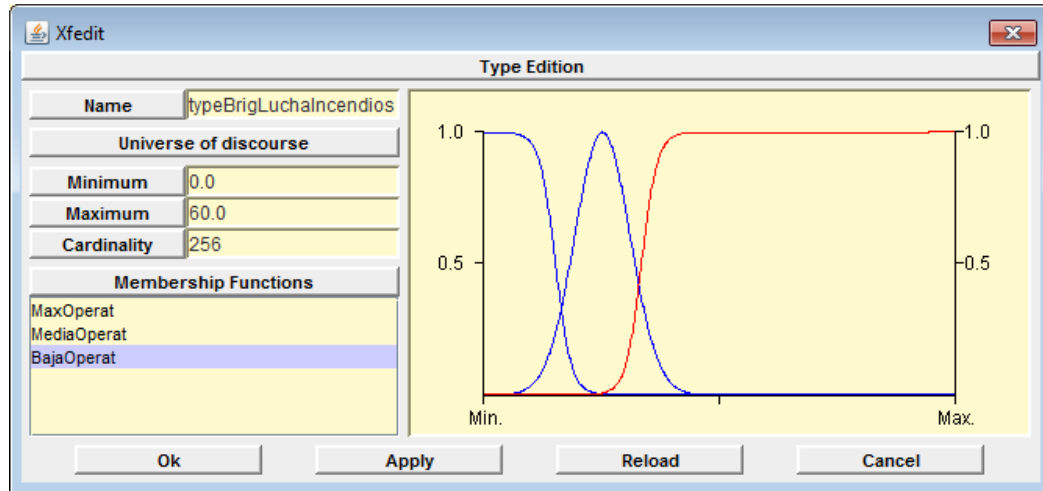


Figura 2.6

✓ **inBomb:** Se refiere a los Bomberos y su universo de discurso es [0,10] (se expresa en kilómetros). A continuación se muestran las funciones de pertenencia en la figura 2.7 donde **Máxima Operatividad** se representa por una función sigma y los parámetros a y b de la misma se evalúan por 2.9 y -0.2; en el caso de **Media Operatividad** se representa por la función campana con los valores de a y b de 4.0 y 1.0 respectivamente y en el caso de **Baja Operatividad** es una sigma cuyos parámetros se evalúan con los valores 5.0 y 0.2 respectivamente:

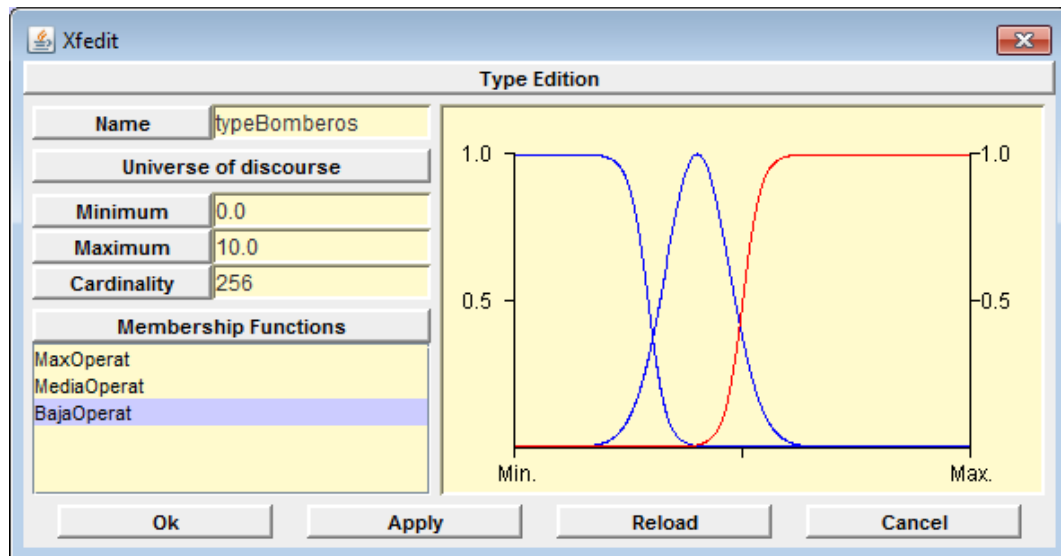


Figura 2.7

- ✓ **inAmbul:** Se refiere a las Ambulancias y su universo de discurso es $[0,10]$ (se expresa en kilómetros) y se modela de forma análoga a **inBomb**.
- ✓ **inPAYO:** Se refiere a los Primeros Auxilios y Otros, su universo de discurso es $[0,60]$ (se expresa en minutos) y se modela de forma análoga a **inBLI**.
- 3. **inSHU:** Se refiere a los Servicios Hospitalarios de Urgencias y su universo de discurso es $[0,10]$ (se expresa en kilómetros) y se modela de forma análoga a **inBomb**.
- 4. **inRB:** Se refiere a la Recuperación Básica y su universo de discurso va de 0 a 1 y se modela de forma análoga a **outAI**.
- 5. **outNivDH:** Se refiere al Nivel de Desarrollo Humano, su universo de discurso es $[0,1]$ y se modela de forma análoga a **outAI**, esta depende de:
 - ✓ **inLong:** Se refiere a la Longevidad y su universo de discurso es $[0,100]$ (se expresa en años).
 - ✓ **inEscol:** Se refiere a la Escolaridad, su universo de discurso es $[0,1]$ y se modela de forma análoga a **outAI**.
 - ✓ **inIPC:** Se refiere al Ingreso Per Cápita y su universo de discurso es $[0,50]$ (se expresa en CUP por día).

Las variables **inLong** y **inEscol** se modelan con los términos lingüísticos *Alta*, *Media*, *Baja*, mientras que la última **inIPC** con *Bien*, *Regular*, *Mal*.
- 6. **inRConst:** Se refiere a la Reconstrucción y su universo de discurso es $[0,1]$.

Las variables **inSS**, **outOE**, **inSHU**, **inRB** y **inRConst** se modelan con los términos lingüísticos *Bien*, *Regular*, *Mal* excepto **outNivDH** que se modela con *Alto*, *Medio*, *Bajo*.

En el anexo II aparecen las funciones de pertenencia asociadas a estas 6 variables.

Variable Vulnerabilidad:

Se definieron cuatro SIB diferentes en dependencia del evento: incendio, explosión, derrame y escape; ya que a pesar de que todos dependen de las mismas variables, estas varían su importancia relativa según el evento y por lo tanto cambian las reglas de inferencia.

- **outVulnerabilidad:** es la variable de salida, su universo de discurso es [0,1] y los términos lingüísticos son: *Muy Baja*, *Baja*, *Media*, *Alta*, *Muy Alta* y se utiliza como método de defuzzificación la media de los máximos, depende de las siguientes variables:
 - ✓ **inGravIndiceFallec:** se refiere al índice de gravedad de fallecidos.
 - ✓ **inGravIndiceLesion:** se refiere al índice de gravedad de lesionados.
 - ✓ **inDInterMayores:** se refiere a los daños internos mayores.
 - ✓ **inDExterMayores:** se refiere a los daños externos mayores.
 - ✓ **inDMedioambiental:** se refiere a los daños medioambientales.

Estas cinco variables se modelan con los términos lingüísticos **Bajo**, **Medio**, **Alto** y su universo de discurso es [0,1].

Las funciones de pertenencia se construyeron utilizando también las funciones sigmas y campanas y tienen la siguiente forma para las cinco variables de entrada:

Para **Bajo** se define una función sigma con los parámetros a y b con los valores 0.25 y 0.01 respectivamente; en el caso de **Medio** a la función campana se les dio los valores de a y b de 0.5 y 0.2 respectivamente y en el caso de **Alto** que es una sigma estos parámetros toman los valores 0.75 y 0.01 respectivamente. La figura 2.8 representan las funciones de pertenencia de la **inGravIndiceFallec** el rojo representa al término **Bajo**:

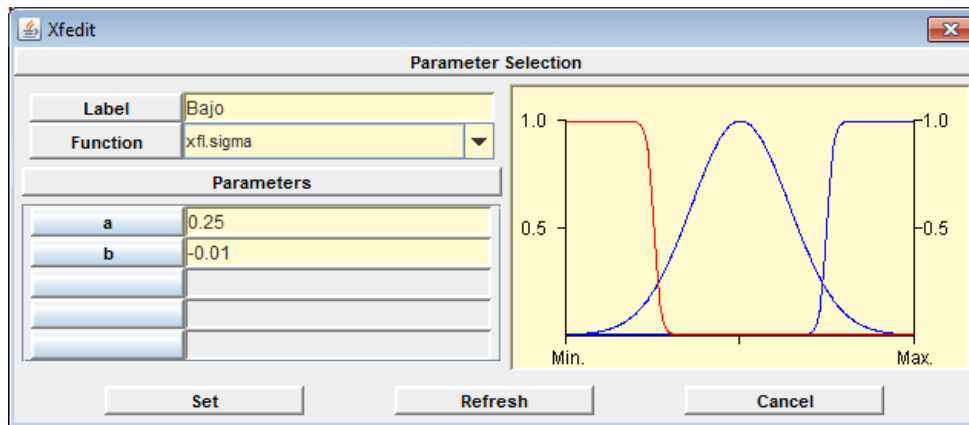


Figura 2.8

De esta forma quedaron constituidos los cuatro SIB: **VulnerabilidadExplosión**, **VulnerabilidadDerrame**, **VulnerabilidadEscape** y **VulnerabilidadIncendio** referentes a cada desastre.

Variable Frecuencia:

- **outFrec:** describe la Frecuencia de aparición de un riesgo, tiene un universo de discurso $[0,1]$, utiliza como método de defuzzificación el máximo de los máximos y se toman como términos lingüísticos *Frecuente*, *Posible*, *Raro*, *Extremadamente Raro* e *Improbable*.

La variable **outFrec** se modela utilizando funciones sigma y campana. Para lograr esto se les dio los valores de a y b en la función sigma de 1.0 y 0.15 respectivamente en el caso de *Frecuente*. En el caso de *Posible* se utilizó una función campana con valores de 0.7 y 0.1 para a y b respectivamente. Para *Raro*, campana también, los valores son de 0.5 y 0.1 para a y b respectivamente. En el caso de *Extremadamente Raro* que igualmente es campana los valores de a y b son de 0.3 y 0.1 respectivamente. Para *Improbable* que es sigma los valores de a y b son de 0.15 y -0.015 respectivamente. A continuación se muestra la figura 2.9 donde se representa la modelación borrosa de la variable **outFrec**.

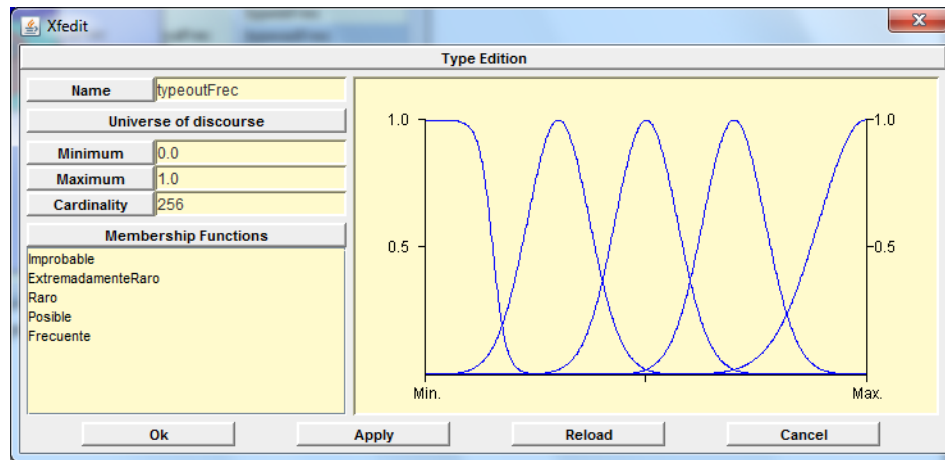


Figura 2.9

La variable **outFrec** se determina a partir de la variable **inFrec** la cual describe el tiempo de ocurrencia del fenómeno y tiene como universo de discurso [0,150] representando meses, **inFrec** se modela también utilizando términos lingüísticos, donde:

- ✓ **Frecuente**: menos de un mes.
- ✓ **Posible**: de un mes a un año.
- ✓ **Raro**: de un año a cinco años.
- ✓ **Extremadamente raro**: de cinco años a diez años.
- ✓ **Improbable**: Más de diez años.

Para lograr esto se les dio los valores de a y b en la función sigma de 1.0 y -0.1 respectivamente en el caso de **Frecuente**. En el caso de **Posible** se utilizó una función campana con valores de 8.0 y 3.0 para a y b respectivamente. Para **Raro**, campana también, los valores son de 30.0 y 10.0 para a y b respectivamente. En el caso de **Extremadamente Raro** que igualmente es campana los valores de a y b son de 80.0 y 25.0 respectivamente. Para **Improbable** que es sigma los valores de a y b son de 115.0 y 3.0 respectivamente. La figura 2.10 representa la modelación borrosa de la variable **inFrec**.

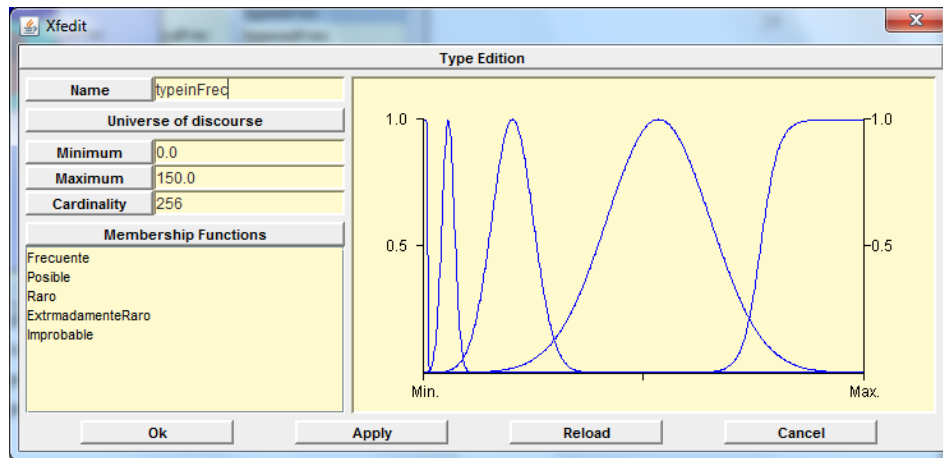


Figura 2.10

2.3 Modelación de los distintos SIB que constituyen el IRT

Se desarrollaron seis SIB: **AgravamientoImpacto**, **VulnerabilidadIncendio**, **VulnerabilidadExplosión**, **VulnerabilidadDerrame**, **VulnerabilidadEscape** y **Frecuencia**. A continuación detallaremos cada uno de ellos:

2.3.1 SIB AgravamientoImpacto

En el caso del SIB **AgravamientoImpacto**, referente a la variable **CAI**, se decidió seleccionar el modelo Mamdani ya que necesitamos utilizar un modelo donde el antecedente y el consecuente de las reglas sean conjuntos borrosos, ya que todas nuestras variables (tanto las de entrada de nuestro sistema como las de salida) son borrosas; y que las reglas puedan tener múltiples antecedentes, debido a que en nuestro sistema existen variables que dependen de otras múltiples variables. Como se mencionó anteriormente se seleccionó como método de defuzzificación la media de los máximos.

Se construyeron ocho bases de reglas de inferencia borrosas constituyendo un sistema jerárquico donde el consecuente de unas era antecedente de otras, el cual se puede apreciar en la figura 2.11:

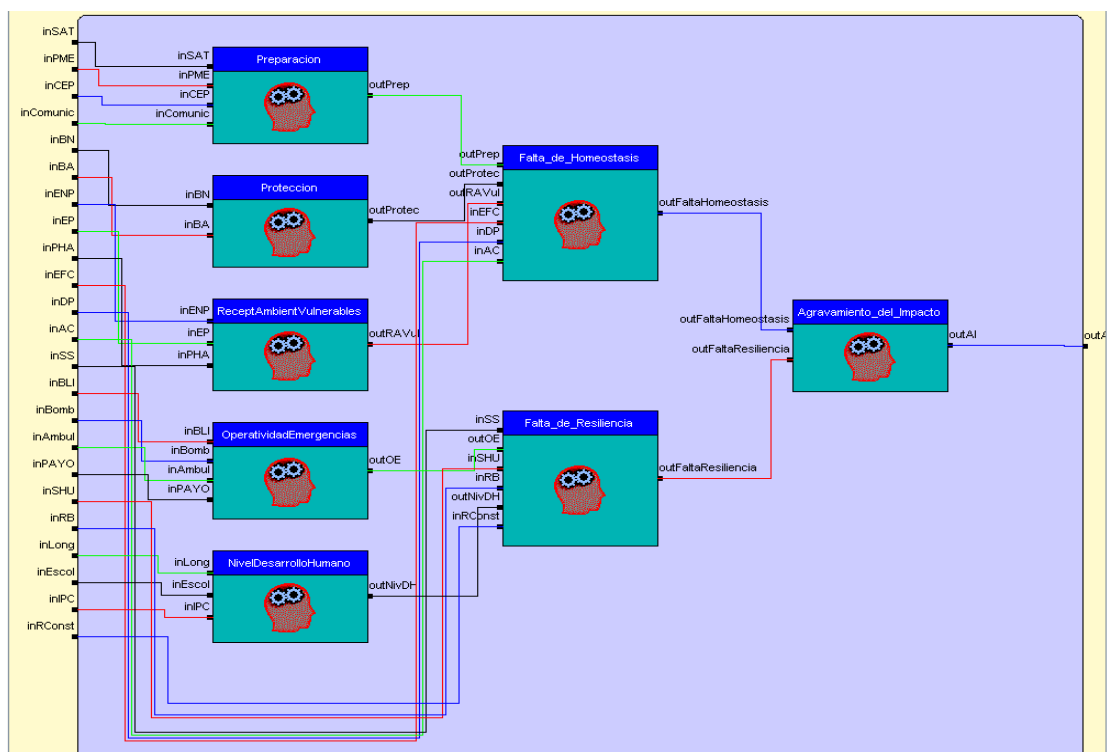


Figura 2.11

Para construir la misma se tuvo en cuenta la importancia relativa de las variables, obtenidas a partir de un análisis estadístico de estas realizado por el experto, estas se muestran en la tabla 2.1:

	Descriptor	Vector propio principal (V)	Vector prioritario	PESOS %
Falta de Homeostasis	Densidad de población	0,6780	0,25	25
	Área comunitaria	0,3958	0,14	14
	Preparación	0,3497	0,13	13
	Protección	0,2516	0,09	9
	Receptores ambientales vulnerables	0,2824	0,10	10
	Estado físico de las construcciones	0,2337	0,08	8
Falta de Resiliencia	Servicios de salvamento	0,1540	0,06	6
	Operatividad en las emergencias	0,1471	0,05	5
	Servicios hospitalarios	0,0965	0,04	4
	Recuperación básica	0,0507	0,02	2
	Nivel de desarrollo humano	0,0617	0,02	2
	Reconstrucción	0,0511	0,02	2

Tabla 2.1

A continuación se muestra cómo se definieron las reglas borrosas, según el lenguaje *XFL3* del *XFuzzy* para la base de reglas **Agravamiento_del_Impacto** que es la última y brinda la salida del SIB:

1. `if(outFaltaHomeostasis == Mal) -> outAI = Mal;`
2. `if(outFaltaHomeostasis == Regular & outFaltaResiliencia == Mal) -> outAI = Mal;`
3. `if(outFaltaHomeostasis == Regular & outFaltaResiliencia == Regular) -> outAI = Regular;`
4. `if(outFaltaHomeostasis == Regular & outFaltaResiliencia == Bien) -> outAI = Regular;`
5. `if(outFaltaHomeostasis == Bien & outFaltaResiliencia == Mal) -> outAI = Regular;`
6. `if(outFaltaHomeostasis == Bien & outFaltaResiliencia == Regular) -> outAI = Bien;`
7. `if(outFaltaHomeostasis == Bien & outFaltaResiliencia == Bien) -> outAI = Bien;`

Utilizando *Xfuzzy* se editaron y ajustaron todas las reglas ver anexo II.

2.3.2 SIB Vulnerabilidad

Como se expresó anteriormente, fue necesario definir cuatro SIB para cada evento y las reglas difusas para cada tipo de vulnerabilidad, debido a que la importancia relativa de las variables de las cuales depende esta, cambian según el tipo de evento, por lo tanto las reglas de inferencia son distintas en cada sistema.

Cada uno de ellos conforma un sistema inferencia tipo Mamdani y utiliza como método de defuzificación la media de los máximos.

Los cuatro sistemas poseen una única base de reglas **Rule_Vulner**.

Para construir las reglas se tuvo en cuenta la importancia relativa de las variables según el tipo de evento, las cuales se muestran en las tablas 2.2, 2.3, 2.4 y 2.5:

Vulnerabilidad Escape:

	Vector propio	Vector prioritario	PESOS
Descriptor	principal		%
Gravedad del Índice de Fallecidos	0,8148	0,47	47
Gravedad del Índice de Lesionados	0,5118	0,29	29
Daños internos mayores	0,0917	0,05	5
Daños externos mayores	0,0870	0,05	5

<i>Daños medioambientales</i>	0,2412	0,14	14
--------------------------------------	--------	------	-----------

Tabla 2.2

Vulnerabilidad Derrame:

Descriptor	Vector propio principal	Vector prioritario	PESOS %
<i>Gravedad del Índice de Fallecidos</i>	0,710	0,37	37
<i>Gravedad del Índice de Lesionados</i>	0,573	0,30	30
<i>Daños internos mayores</i>	0,155	0,08	8
<i>Daños externos mayores</i>	0,117	0,06	6
<i>Daños medioambientales</i>	0,361	0,19	19

Tabla 2.3

Vulnerabilidad Explosión:

Descriptor	Vector propio principal	Vector prioritario	PESOS %
<i>Gravedad del Índice de Fallecidos</i>	0,8293	0,47	47
<i>Gravedad del Índice de Lesionados</i>	0,4794	0,27	27
<i>Daños internos mayores</i>	0,2428	0,14	14
<i>Daños externos mayores</i>	0,1325	0,08	8
<i>Daños medioambientales</i>	0,0775	0,04	4

Tabla 2.4

Vulnerabilidad Incendio:

Descriptor	Vector propio principal	Vector prioritario	PESOS %
<i>Gravedad del Índice de Fallecidos</i>	0,8582	0,50	50
<i>Gravedad del Índice de Lesionados</i>	0,4456	0,26	26
<i>Daños internos mayores</i>	0,2045	0,12	12
<i>Daños externos mayores</i>	0,1176	0,07	7
<i>Daños medioambientales</i>	0,0964	0,06	6

Tabla 2.5

A continuación se muestra el ejemplo de la base de reglas para el SIB **VulnerabilidadIncendio**, para el resto ver anexo II:

1. `if(inGravIndiceFallec == Alto & inGravIndiceLesion == Bajo & inDInterMayores == Bajo & inDExterMayores == Bajo & inDMedioambiental == Bajo) -> outVulnerabilidad = Media;`
2. `if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Bajo & inDInterMayores == Bajo & inDExterMayores == Bajo & inDMedioambiental == Bajo) -> outVulnerabilidad = MuyBaja;`
3. `if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio & inDInterMayores == Alto & inDExterMayores == Alto & inDMedioambiental == Alto) -> outVulnerabilidad = MuyAlta;`
4. `if(inGravIndiceFallec == Medio & inGravIndiceLesion == Alto) -> outVulnerabilidad = MuyAlta;`
5. `if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio & inDInterMayores == Medio & inDExterMayores == Alto & inDMedioambiental == Alto) -> outVulnerabilidad = Alta;`
6. `if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio & inDInterMayores == Medio & inDExterMayores == Medio & inDMedioambiental == Medio) -> outVulnerabilidad = Media;`
7. `if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Bajo & inDInterMayores == Medio) -> outVulnerabilidad = Baja;`
8. `if(inGravIndiceFallec == Medio & inGravIndiceLesion == Bajo & inDInterMayores == Bajo & inDExterMayores == Bajo & inDMedioambiental == Bajo) -> outVulnerabilidad = Media;`
9. `if(inGravIndiceFallec == Alto & inGravIndiceLesion == Alto & inDInterMayores == Alto & inDExterMayores == Alto & inDMedioambiental == Alto) -> outVulnerabilidad = MuyAlta;`

2.3.3 SIB Frecuencia

El SIB Frecuencia se implementó de forma similar a los anteriores sistemas. Este sistema es también de tipo Mamdani, ya que tanto la variable de entrada como la de salida son borrosas. El método de defuzzificación escogido fue el máximo de los máximos debido a que fue el que mejores resultados arrojó.

La construcción de las reglas borrosas se realizó a partir de los requerimientos del experto y el sistema posee una única base de regla que se muestra a continuación:

1. `if (inFrec == Frecuente) -> outFrec = Frecuente;`
2. `if (inFrec == Posible) -> outFrec = Posible;`
3. `if (inFrec == Raro) -> outFrec = Raro;`
4. `if (inFrec == ExtrmadamenteRaro) -> outFrec = ExtremadamenteRaro;`
5. `if (inFrec == Improbable) -> outFrec = Improbable;`

2.4 Implementación de la interfaz en Java. SIDCIRT como módulo del software SEVAT

Una vez que se tienen definidos los SIB en el Xfuzzy, se hace necesaria una aplicación que utilizando los valores defuzzificados de los mismos, calcule el Índice de Riesgo

Tecnológico (IRT) a partir de la fórmula (I) propuesta en el capítulo 2, además de brindarle al usuario una interfaz amigable de fácil utilización y comprensión para la entrada de valores y la ejecución del cálculo.

Para nuestra aplicación, la cual se decidió llamar SIDCIRT (Sistema Inteligente Difuso para el Cálculo del Índice de Riesgo Tecnológico), se eligió como lenguaje de programación el Java, el cual es un lenguaje de alto nivel orientado a objetos que ofrece amplias facilidades para el desarrollo de aplicaciones y cumple con las características necesarias para brindar una mejor solución a nuestro problema. Se eligió también como Entorno de Desarrollo Integrado (IDE, de sus siglas en inglés) el NetBeans 7.2.1 con JDK 1.7.0_04.

SIDCIRT constituirá un módulo del software SEVAT (Sistema para la evaluación de la vulnerabilidad territorial ante accidentes tecnológicos graves), ya validado y en proceso de evaluación por los expertos. SEVAT es una aplicación de escritorio que brinda las vistas necesarias para la entrada de datos en el cálculo del IRT de las diferentes industrias enmarcadas dentro de un área, los cuales serán almacenados, además la misma realiza consultas sobre las sustancias que componen la producción de dichas industrias en la base de datos, (Medina, 2014). Luego de aplicar a los datos, el modelo para el cálculo de índice (de lo cual se encarga SIDCIRT), se podrá consultar gráficos con ordenamiento en cuanto a la vulnerabilidad de las diferentes áreas, así como la visualización geográfica de las mismas con coloraciones según su grado de riesgo, facilitando de esta manera la toma de decisiones.

2.4.1 Creación del proyecto en Java e integración del XFuzzy

Como se mencionó en el capítulo anterior, el XFuzzy brinda la facilidad de trasladar el archivo resultante de la definición del SIB, el cual es de extensión *.xfl* y con un lenguaje llamado *XFL3*, a código Java. Para esto, por cada SIB, el XFuzzy genera cuatro clases o archivos *.java*, las tres primeras corresponden a descripciones de dos interfaces y una clase que son comunes a todos los SIB, la última contiene la descripción específica del sistema difuso *<nombre_del_SIB>.xfl* y serán explicadas brevemente:

- *FuzzyInferenceEngine.java* describe una interfaz Java que define un sistema de inferencia difuso general, la misma define cuatro métodos para implementar el proceso de inferencia con valores crisp y difusos:

```
public interface FuzzyInferenceEngine {
    public double[] crispInference(double[] input);
    public double[] crispInference(MembershipFunction[] input);
    public MembershipFunction[] fuzzyInference(double[] input);
    public MembershipFunction[] fuzzyInference(MembershipFunction[]
input);
}
```

- *MembershipFunction.java* contiene la descripción de una interfaz usada para describir un número difuso, contiene sólo un método llamado `compute` que calcula el grado de pertenencia para cada valor del universo de discurso del número difuso:

```
public interface MembershipFunction {
    public double compute(double x);
}
```

- *FuzzySingleton.java* implementa la interfaz *MembershipFunction*, representa un valor crisp como un número difuso:

```
public class FuzzySingleton implements MembershipFunction {
    private double value;

    public FuzzySingleton(double value) { this.value = value; }
    public double getValue() { return this.value; }
    public double compute(double x) { return (x==value? 1.0: 0.0); }
}
```

- *<nombre_del_SIB>.java* contiene la clase que describe el sistema difuso. Esta clase es una implementación de la interfaz *FuzzyInferenceEngine*, por tanto, los métodos públicos que implementan la inferencia son los de la interfaz: `crispInference` y `fuzzyInference`.

En el IDE NetBeans se creó un proyecto de igual nombre que la aplicación y se utilizaron cuatro paquetes donde están agrupadas las distintas clases del sistema:

- **gui**: contiene las clases que manejan la interfaz gráfica del sistema.
- **io**: contiene las clases que controlan la entrada y salida del sistema, para la escritura y lectura de fichero.
- **model**: contiene las clases que modelan la lógica de la aplicación.
- **xfuzzy**: contiene las clases generadas por el Xfuzzy que se utilizan en el cálculo del IRT.

2.4.2 Diagrama de clases

En el diagrama de clases del paquete **xfuzzy** que se muestra en la figura 2.12 sólo aparecen las relaciones de las clases, en el anexo III se muestra el diagrama completo.

En la figura 2.13 se muestra el diagrama de las clases principales referentes a la interfaz visual y a la lógica del programa, para el diagrama completo y para el resto de las clases ver anexo III.

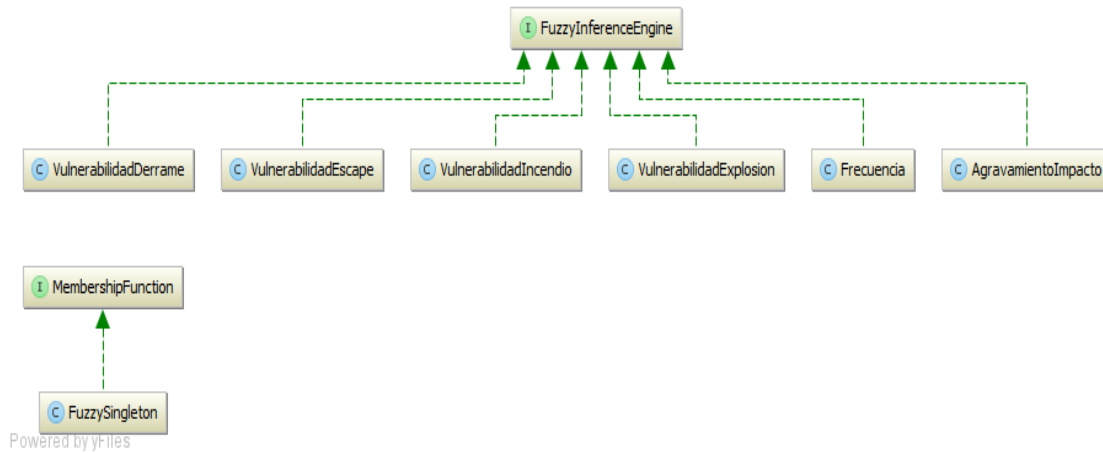


Figura 2.12

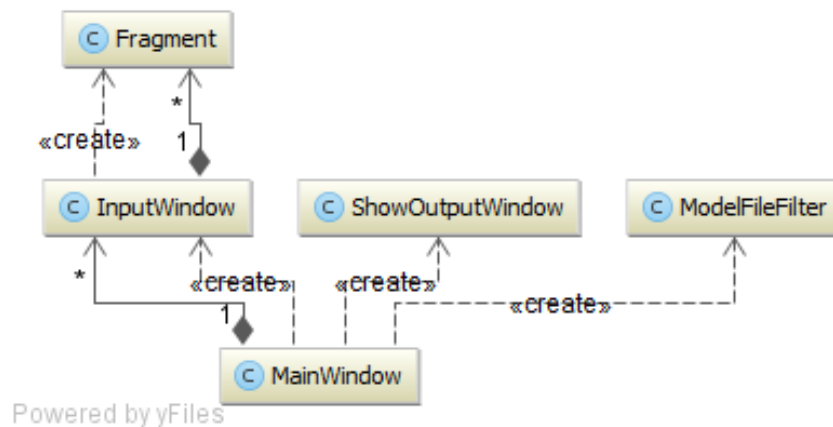


Figura 2.13

2.4.3 Diagrama de componentes

En la figura 2.14 se muestra la relación de los distintos componentes de la aplicación. Como componente principal se tiene el archivo ejecutable *SIDCIRT.jar*, el cual va a depender de las clases generadas por el Xfuzzy, la interfaz de usuario, las clases de entrada y salida, los archivos *XML* de configuración y la biblioteca *JDOM* para el manejo de los *XML*.

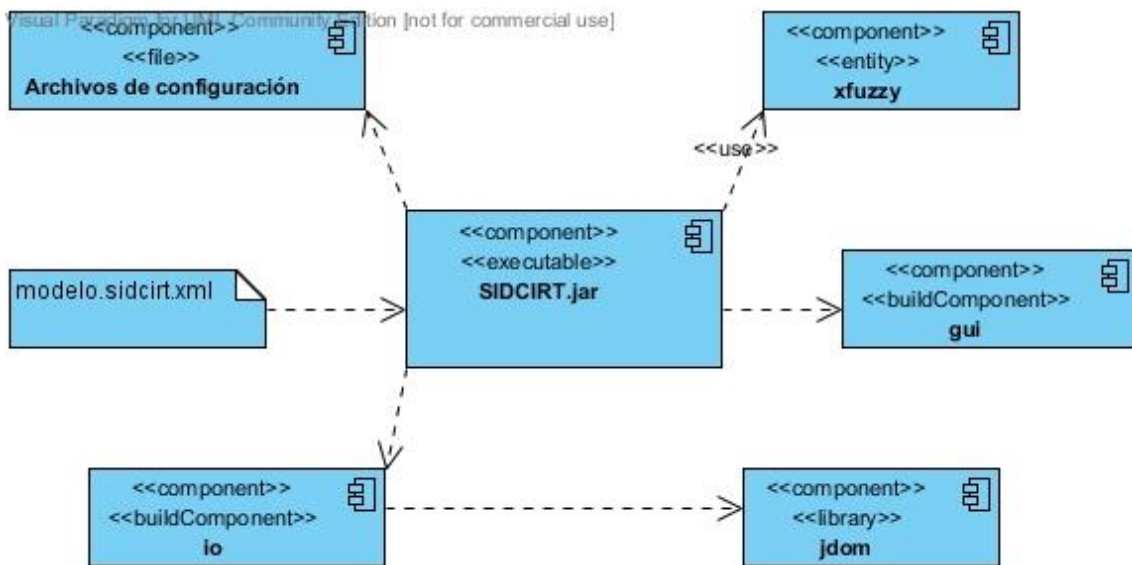


Figura 2.14

2.4.4 Diagrama de actividades

Nuestra aplicación comienza su funcionamiento a partir de la ejecución de SIDCIRT.jar, ver figura 2.15, posteriormente el usuario tiene la posibilidad de introducir manualmente los valores de las distintas variables, que por defecto aparecen en su valor medio o cargar los datos desde un fichero con los valores previamente editados. Luego ejecuta el cálculo del IRT (este desarrolla todo un proceso transparente al usuario, que será expuesto en el diagrama de clases de la figura 2.16), que muestra la evaluación del riesgo. Por último se pueden guardar los datos editados en un fichero.

Visual Paradigm for UML Community Edition [not for commercial use]

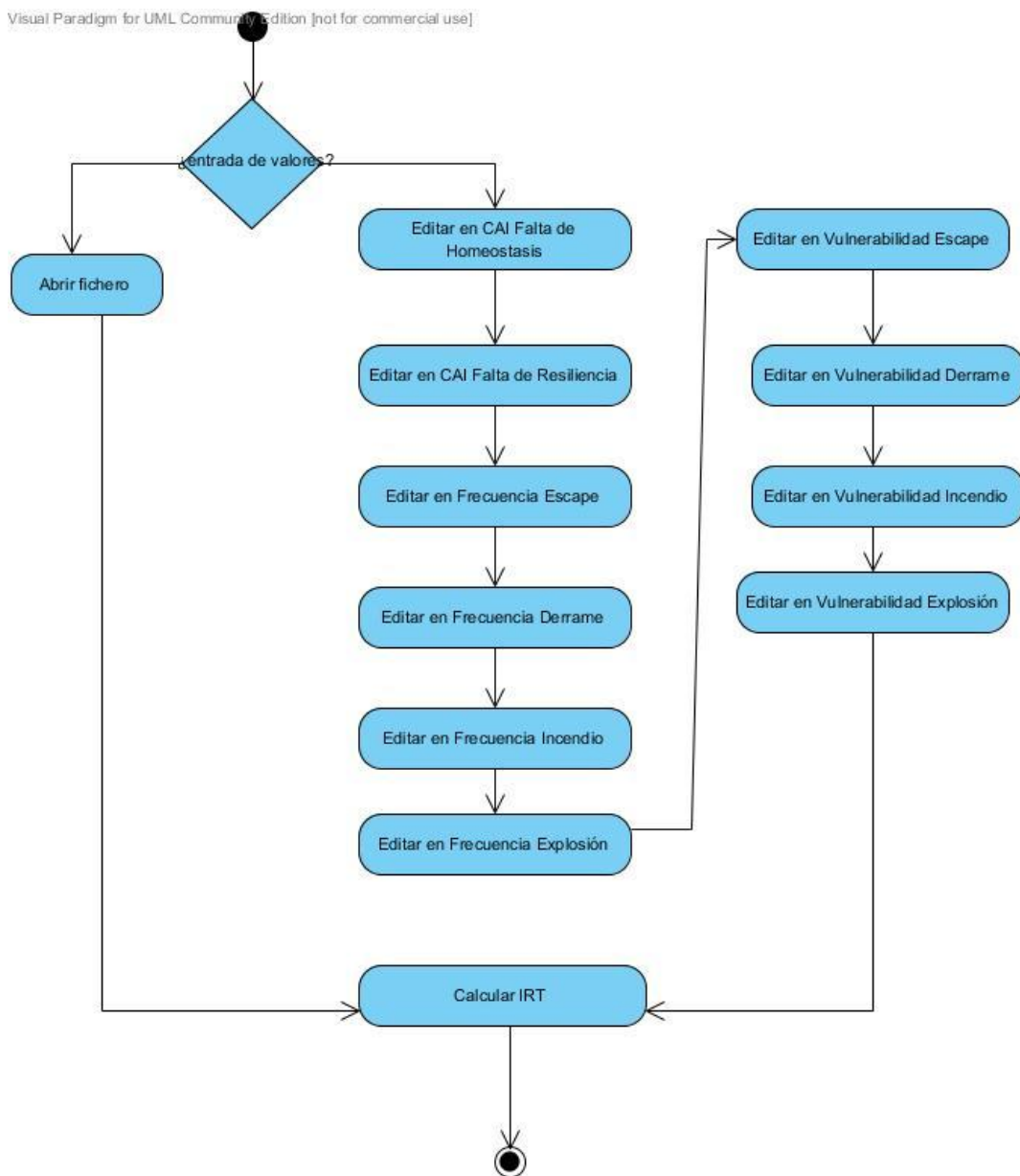


Figura 2.15

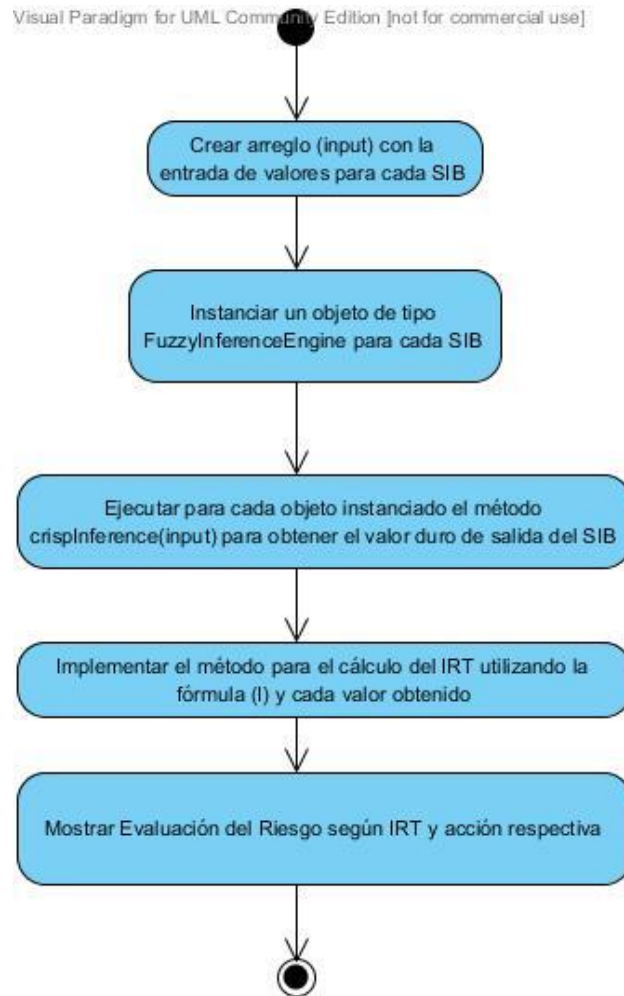


Figura 2.16

2.5 Conclusiones parciales

- El software Xfuzzy3.0 constituye una herramienta útil para la definición de SIB de avanzada complejidad, para el proceso de ajuste de parámetros y la selección de la mejor variante de defuzzificación, lo cual favoreció el desarrollo de los SIB para la determinación del IRT.
- La ingeniería de conocimiento desarrollada permitió que se modelaron en total seis SIB tipo Mamdani: uno para determinar el coeficiente del agravamiento del impacto (CAI), otro por la frecuencia del evento y los demás por la vulnerabilidad física del área según los cuatro posibles eventos; de los cuales se utilizarán sus respectivos valores defuzzificados para el cálculo del IRT.
- SIDCIRT, escrita en Java, es la encargada de integrar todos los SIB modelados por el Xfuzzy para el proceso del cálculo, además de brindarle al usuario una interfaz

amigable de fácil utilización, la cual es fácil de integrar como un módulo del software SEVAT, el cual almacena toda la información necesaria de las diferentes áreas y posee una serie de herramientas que facilitan al proceso de toma de decisiones.

CAPÍTULO 3: SISTEMA INTELIGENTE DIFUSO PARA EL CÁLCULO DEL RIESGO TECNOLÓGICO (SIDCIRT)

En el presente capítulo se describen los requisitos de hardware y software del software SIDCIRT para un óptimo funcionamiento, la evaluación del mismo por parte de los expertos y cómo se integrará al software SEVAT. Para finalizar se incluye un breve manual de usuario con el objetivo de facilitar su utilización por el usuario.

3.1 Requerimientos de la aplicación

La aplicación, se desarrolló en Java, el cual es multiplataforma y demanda un mínimo de requisitos técnicos que se mencionan a continuación:

- Sistemas operativos (cualquiera que soporte un *JRE*), ejemplo:
 - Windows: NT, 2000, XP, Vista, 7, 8.
 - Linux.
 - MacOS.
- Espacio de HD disponible: 5 MB.
- Memoria RAM: mínimo recomendado 128 MB.
- JDK versión: 1.7.0_04

3.2 Validación de SIDCIRT

Para la validación de SIDCIRT se analizaron tres casos de estudio, los cuales se presentan íntegramente en el anexo IV.

El primer caso de estudio presenta una entrada de variables con valores medios, mientras que el segundo contiene valores negativos y el tercer caso contiene valores positivos.

Los resultados son semejantes a los propuestos por el experto, en la tabla 3.1 aparece el resultado generado por el sistema y el evaluado por el experto, pudiéndose apreciar la no existencia de diferencias entre ambos resultados.

Caso de Estudio	Resultado	Evaluación del Experto
1	0.121 - Riesgo Moderado	Riesgo Moderado
2	0.727 - Riesgo Intolerable	Riesgo Intolerable
3	0.001 - Riesgo Trivial	Riesgo Trivial

Tabla 3.1

3.3 Integración con SEVAT

La aplicación SIDCIRT se integra al software SEVAT como un módulo de este.

SEVAT guarda en la base de datos **db_servicio** la información de las áreas de los distintos territorios, con esta información, SIDCIRT procede a realizar el cálculo del IRT de cada área. Inicialmente todo el cálculo del IRT era realizado manualmente para lo cual el usuario editaba cada parámetro requerido.

Para la integración de SIDCIRT a SEVAT se deben incluir los paquetes de SIDCIRT: **model**, **io** y **xfuzzy**, los cuales se describieron en el capítulo anterior, debido a que la parte de la lógica del programa es independiente de la interfaz.

La entrada de los datos de SIDCIRT que se realiza a partir de un fichero o por edición manual, debe adaptarse para que reciba los valores que se almacenan en la base de datos de SEVAT.

Por otra parte se requieren cambios en la interfaz de la entrada de los datos de las áreas de SEVAT pues se ha cambiado la concepción de algunas variables y se han incluido otras.

3.4 Funcionamiento general de SIDCIRT

La aplicación comienza su funcionamiento con la ejecución del fichero SIDCIRT.jar, seguidamente aparece la ventana principal figura 3.1, la cual contiene una barra menú donde se agrupan las opciones que brinda el sistema:

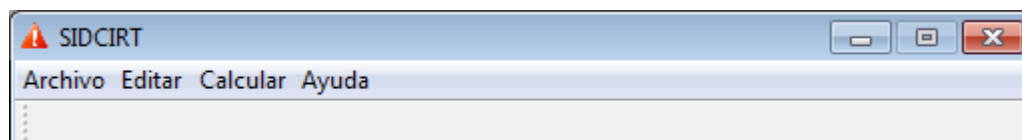


Figura 3.1

- Menú “Archivo”, figura 3.2 contiene las siguientes opciones:

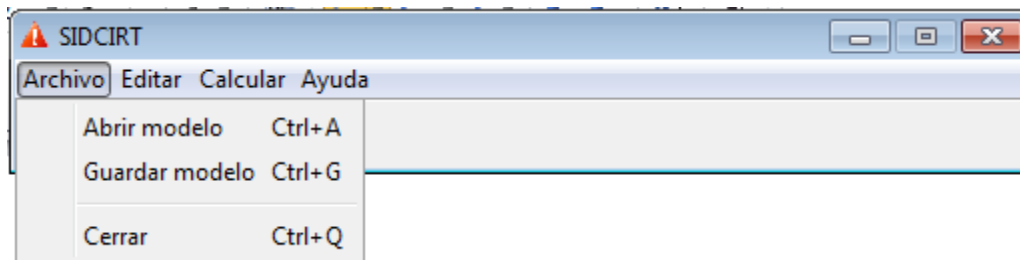


Figura 3.2

- **Abrir modelo:** permite cargar ficheros con la entrada de valores de las variables del sistema ya inicializados, los cuales deben tener la extensión *.sidcirt.xml*.

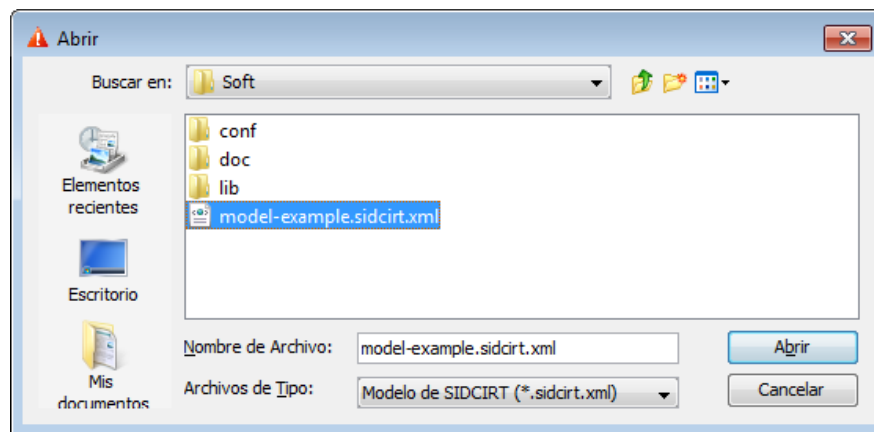


Figura 3.3

- **Guardar modelo:** Permite al usuario guardar los datos de las variables que ha introducido en un fichero de extensión *.sidcirt.xml*., figura 3.4.

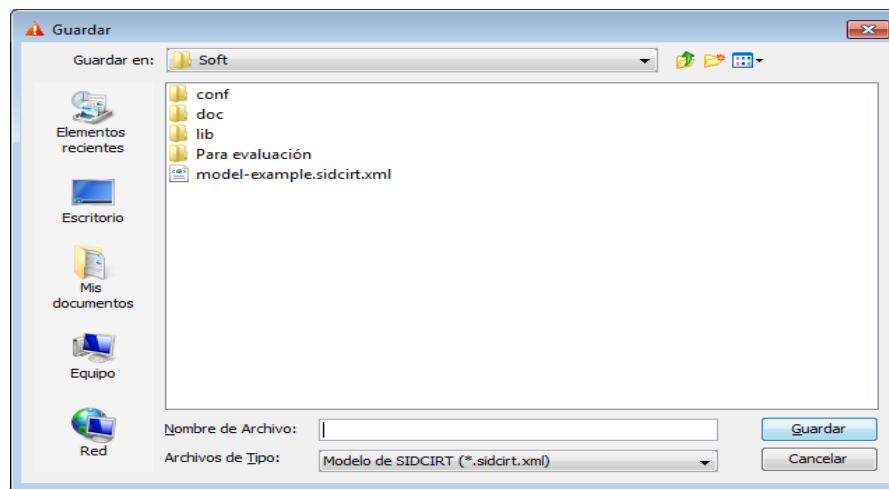


Figura 3.4

- **Cerrar:** Cierra el sistema.
- **Menú “Editar”:** Permite que el usuario entre los valores de las distintas variables necesarias para el cálculo del IRT agrupadas en los submenús “CAI”, “Vulnerabilidad” y “Frecuencia”, atendiendo a los parámetros fundamentales para el cálculo de IRT.
 - **“CAI”:** Permite acceder a la edición de sus dos variables principales **“Falta de Homeostasis”** y **“Falta de Resiliencia”**, figura 3.5:

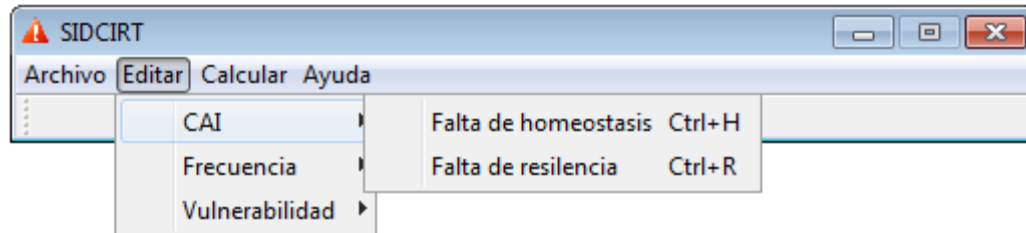


Figura 3.5

- ❖ **“Falta de Homeostasis”:** Permite editar los valores de los parámetros de los cuales depende la homeostasis. En la figura 3.6 se muestra como el usuario puede entrar este valor usando el teclado o auxiliándose de un *slider* que representa el dominio del parámetro.
- ❖ **“Falta de Resiliencia”:** Permite editar los valores de los parámetros de los cuales depende la resiliencia, en forme similar a la homeostasis, la figura 3.7 representa estos.

Figura 3.6

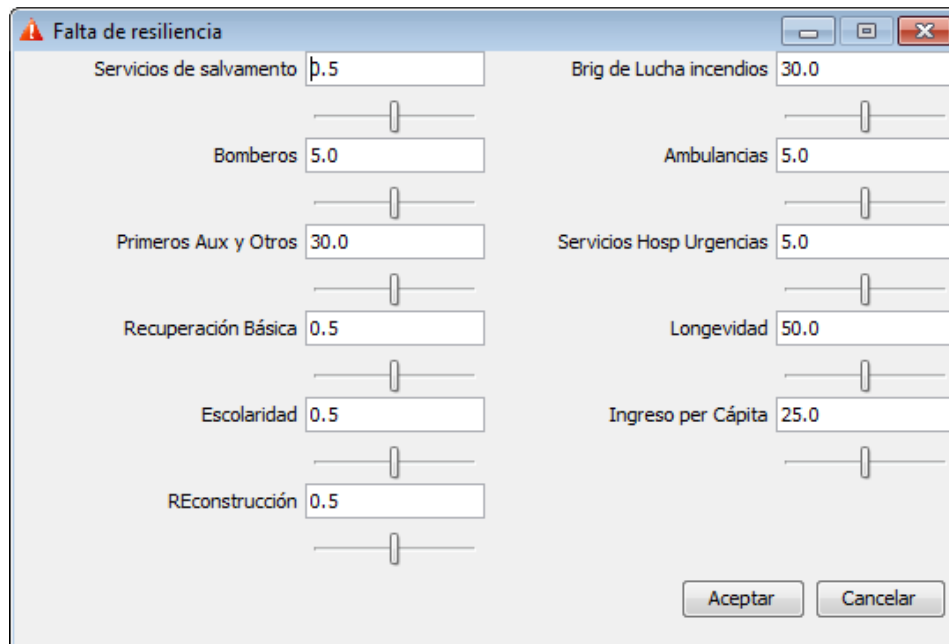


Figura 3.7

- **“Frecuencia”**: Permite la edición de las distintas frecuencias según el evento **“Escape”**, **“Derrame”**, **“Explosión”** e **“Incendio”**, figura 3.8:

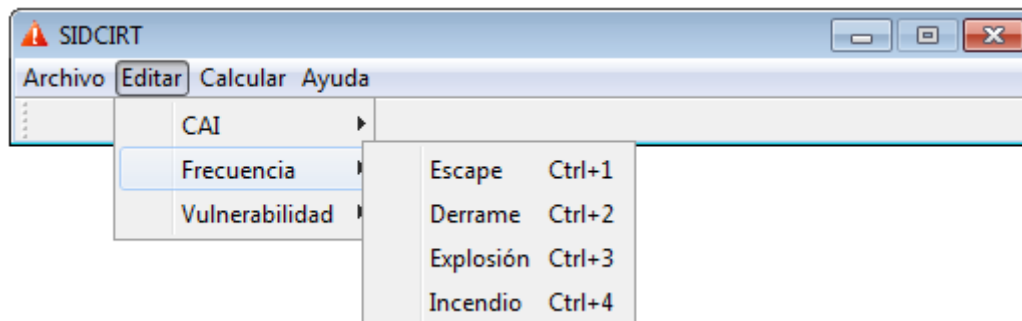


Figura 3.8

- ❖ **“Escape”**: Esta opción permite editar el tiempo en que ocurre un escape, figura 3.9.
- ❖ **“Derrame”**: Esta opción permite editar el tiempo de ocurrencia de derrame.
- ❖ **“Explosión”**: Esta opción permite editar el tiempo de ocurrencia de explosión.
- ❖ **“Incendio”**: Esta opción permite editar el tiempo de ocurrencia de incendio.

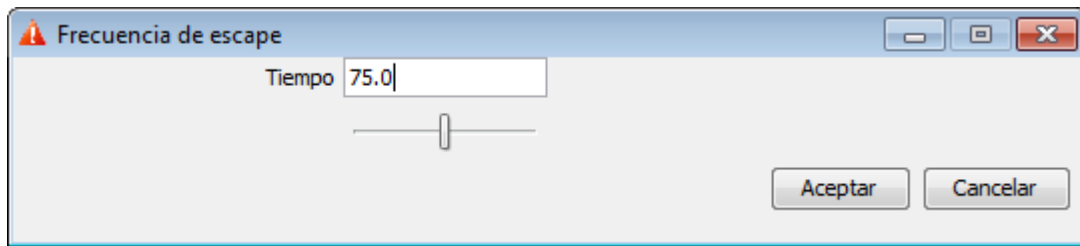


Figura 3.9

- **“Vulnerabilidad”**: Permite la edición los parámetros asociados a las distintas vulnerabilidades según el evento **“Escape”**, **“Derrame”**, **“Explosión”** e **“Incendio”**, figura 3.10:

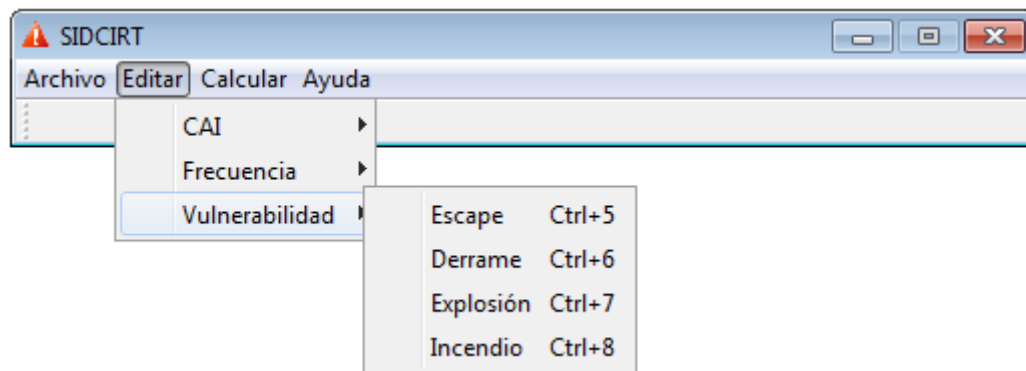


Figura 3.10

- ❖ **“Escape”**: Esta opción permite editar los parámetros asociados a la vulnerabilidad de escape, figura 3.11. En forma similar se proyectan las opciones para derrame, explosión e incendio.

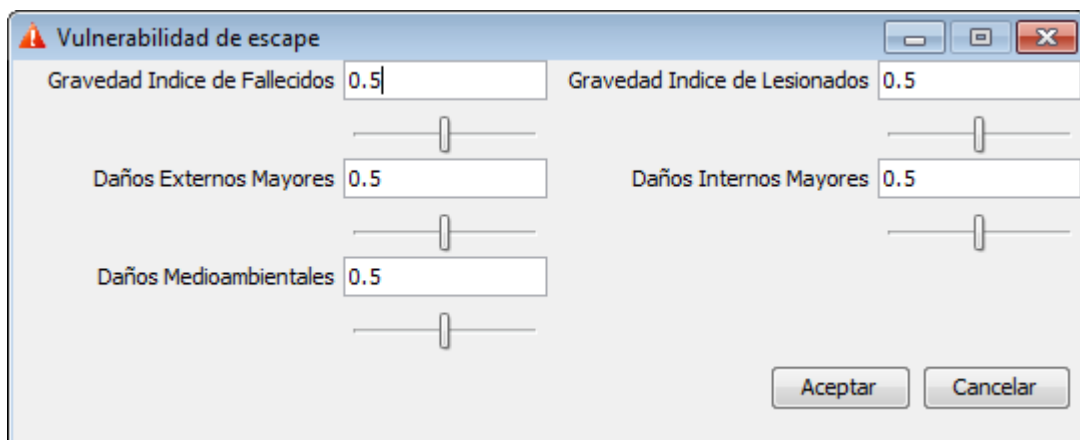


Figura 3.11

- **Menú “Calcular”**: calcula el Índice de Riesgo Tecnológico (IRT), según los valores de los parámetros previos. En la figura 3.12 aparece la ventana con la información relativa a IRT para lo cual aparece el valor calculado del mismo, el término lingüístico donde se alcanza el mayor grado de pertenencia e indica las acciones a seguir para mitigar el riesgo.

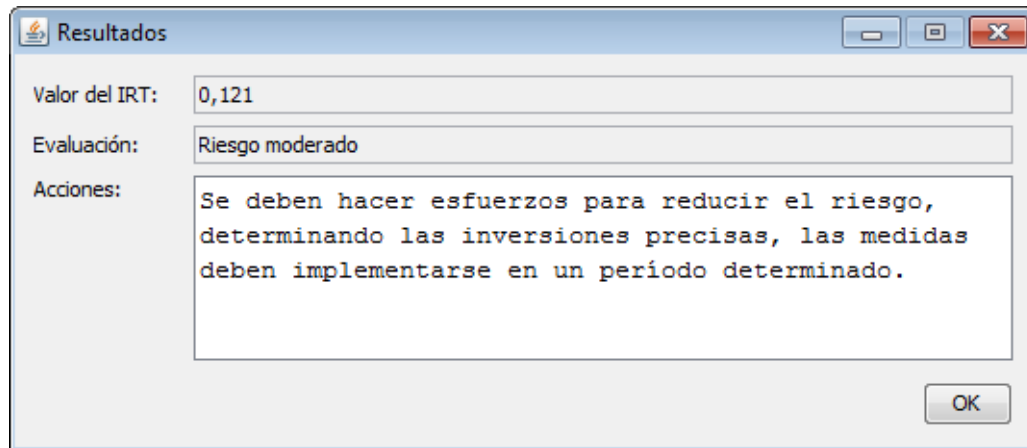


Figura 3.12

- **“Menú Ayuda”**: Muestra las opciones “Ayuda” y “Acerca...”, figura 3.13:

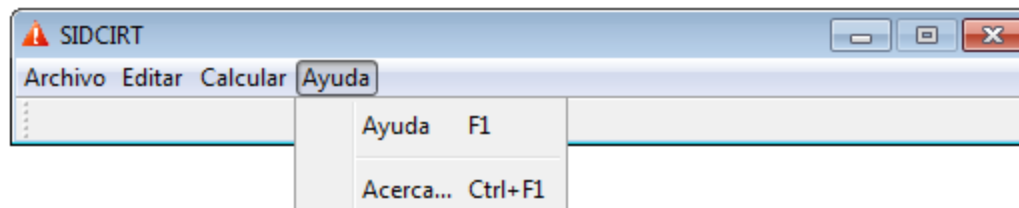


Figura 3.13

- **“Ayuda”**: Permite acceder al manual de usuario de SIDCIRT, guardado en el fichero *help.HTML*, figura 3.12:

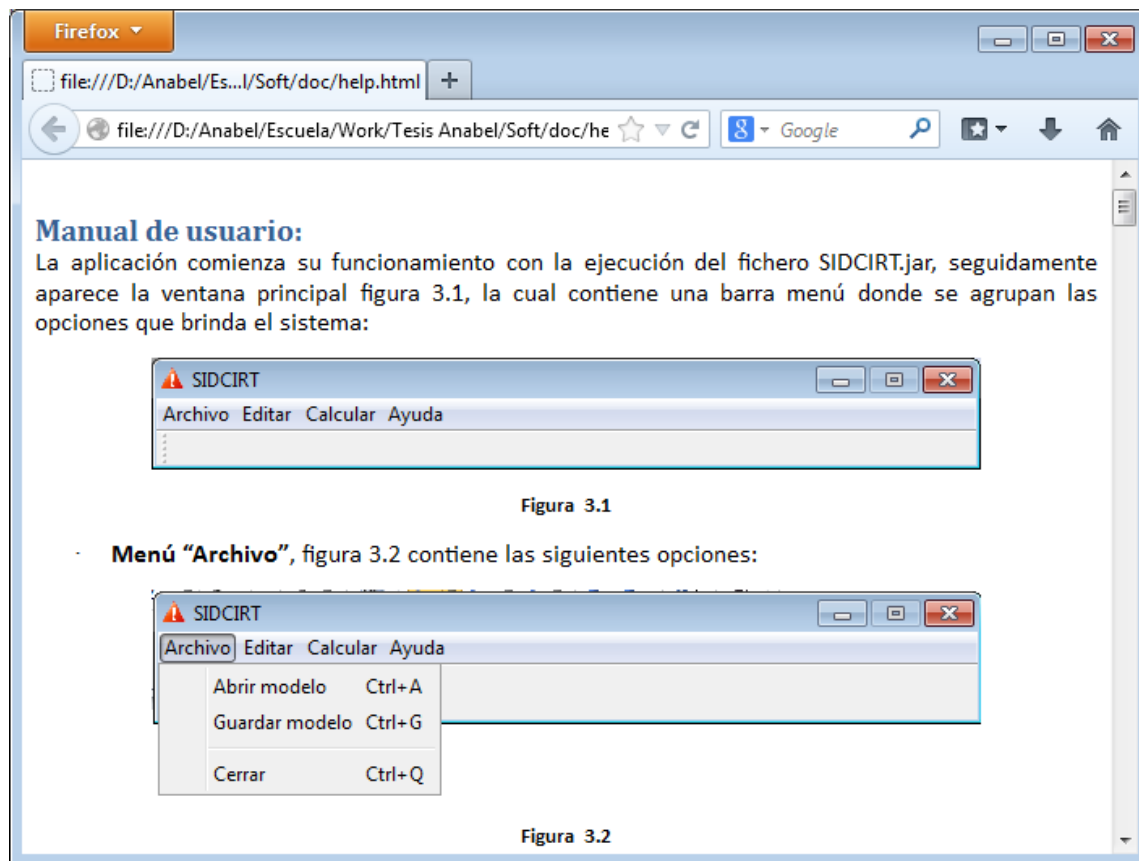


Figura 3.12

- **“Acerca...”:** Muestra una ventana con la información del software, figura 3.13:

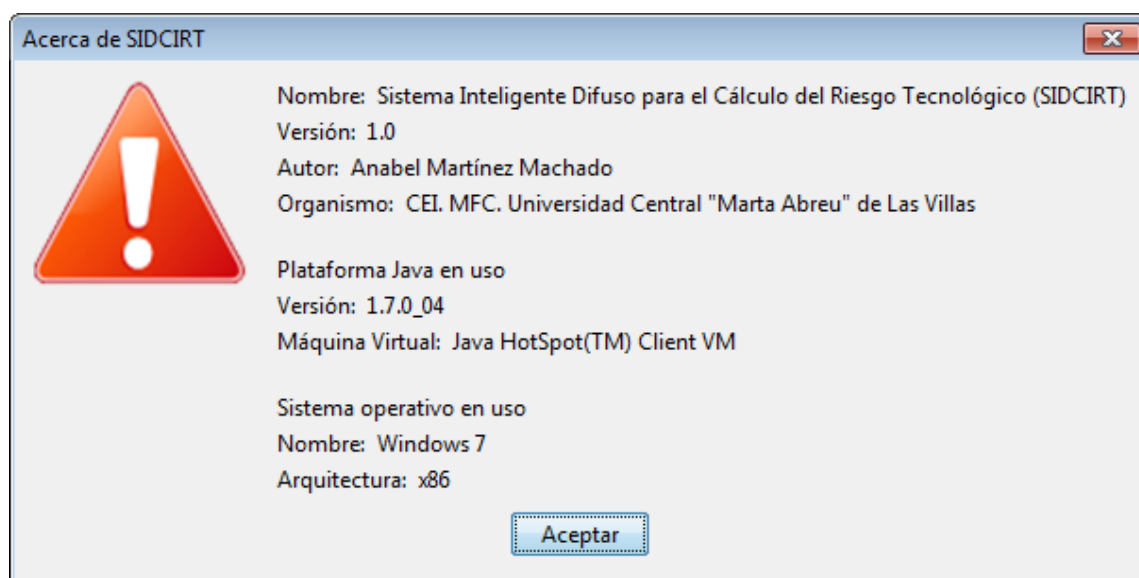


Figura 3.13

3.5 Conclusiones parciales

- SIDCIRT, al ser programado puramente en Java, comparte la característica de ser multiplataforma, por lo tanto funciona en cualquier sistema operativo que soporte un entorno de ejecución de Java o *JRE*, lo cual lo convierte en un software portable. Su interfaz visual resulta amigable y de fácil utilización para cualquier usuario.
- Los casos de estudios desarrollados muestran resultados satisfactorios y han favorecido el ajuste de las funciones de pertenencia, permitiendo validar el sistema.
- La integración de SIDCIRT a SEVAT se reduce a elementos de interfaz y a la captación de 3 módulos correctamente evaluados desde SIDCIRT.

CONCLUSIONES

- Se definieron los factores que caracterizan el coeficiente de agravamiento del impacto (CAI), la vulnerabilidad física del área en estudio y la frecuencia del evento.
- Se realizó la modelación borrosa de las diferentes variables que caracterizan a dichos factores utilizando la herramienta Xfuzzy3.0, a partir de la ingeniería de conocimiento desarrollada, obteniéndose en este proceso seis SIB como resultado de los ajustes en el proceso de validación.
- Se implementó el módulo SIDCIRT, con una interfaz amigable que se integra al software SEVAT, actualmente evaluado a partir de tres casos de estudios por especialistas del dominio.

RECOMENDACIONES

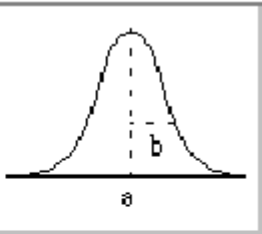
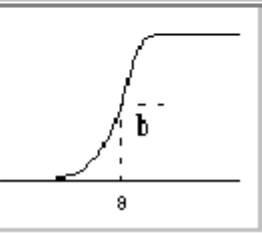
Para el perfeccionamiento de SIDCIRT se recomienda continuar el ajuste de las funciones de pertenencias a partir de las opciones que ofrece Xfuzzy para lo cual se deberán generar ficheros de casos de estudio.

Extender el análisis a otros tipos de amenazas como por ejemplo los desastres naturales.

REFERENCIAS BIBLIOGRÁFICAS

- ARBOLEDA, O. D. C. 2008. Medición de la gestión del riesgo en América Latina. *Revista Internacional de Sostenibilidad, Tecnología y Humanismo*, 3.
- BELLO, R. 1998. *SOLUCION DE PROBLEMAS BAJO INCERTIDUMBRE. SISTEMAS DE INFERENCIA BORROSOS*. Universidad Cooperativa de Colombia. Universidad Central de Las Villas
- CARDONA, O. D. 2001. *Estimación Holística del Riesgo Sísmico utilizando sistemas dinámicos complejos*. Doctoral, Universidad Politecnica de Catalunya. Barcelona.
- CORDÓN, O. Diseño de Sistemas Difusos para Modelado y Clasificación. Aplicaciones. Universidad de Granada.
- COX, E. 1994. *The fuzzy systems handbook*.: Academic Press.
- IMSE-CNM 2003a. FUZZY LOGIC DESIGN TOOLS. IMSE-CNM.
- IMSE-CNM 2003b. *HERRAMIENTAS DE CAD PARA LÓGICA DIFUSA. XFuzzy 3.0*.
- LIO, D. G. 1998. *Curso de Sistemas Basados en el Conocimiento*. , Universidad Central “Martha Abreu” de Las Villas.
- MATA-GARCÍA, M. G., AGUDO, E. T. & RIVAS, F. G. 2006. *SISTEMA DE RIEGO INTELIGENTE BORROSO* UNIVERSIDAD COMPLUTENSE DE MADRID.
- MEDINA, R. C. 2014. *Sistema para la evaluación de la vulnerabilidad territorial ante accidentes tecnológicos graves (SEVAT)*. Maestría, UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS.
- MORELL, C. & BELLO, R. 2007. *Una arquitectura unificada para el razonamiento borroso*.
- NNIGHT, E. R. Y. K. 1991. *Artificial Intelligence*.: McGraw Hill.
- OFFICE, U. N. D. R. 1980. *Natural Disasters and Vulnerability Analysis*
- PIÑERO, P., PÉREZ, A. D., RAMÍREZ, E. C., PIÑERO, P. R. & HERRERA, R. Y. 2006. Un modelo para la ayuda a la toma de decisiones basado en técnicas de SOFTCOMPUTING. Uso Potencial en el diagnóstico médico especializado. *Bioingeniería y Física Médica Cubana*, 7.
- SOCIETY, I. I. A. 2009. *Curso: Aplicaciones Industriales de Fuzzy Logic*.
- TURBAN, E. 1992. *Expert Systems and Applied Artificial Intelligence*.: Macmillan Publishing. .
- VALDIVIA, A. G., IBARRA, V. & GUERRA, B. 2010. Informe de investigación: Índice Global de Riesgo. *In: CEQA, U. (ed.)*.

ANEXO I: Funciones de pertenencia utilizadas y métodos de defuzzificación. Definiciones del XFuzzy 3.0.

Función	Parámetro	Gráfica
bell	a,b	
sigma	a,b	

Bell

```
private class MF_xfl_bell extends InnerMembershipFunction {
    double a;
    double b;

    MF_xfl_bell( double min, double max, double step, double param[]){
        super.min = min;
        super.max = max;
        super.step = step;
        this.a = param[0];
        this.b = param[1];
    }
    double param(int _i) {
        switch(_i) {
            case 0: return a;
            case 1: return b;
            default: return 0;
        }
    }
    double isEqual(double x) {
        return Math.exp( -(a-x)*(a-x)/(b*b) );
    }
    double isGreaterOrEqual(double x) {
        if(x>a) return 1; return Math.exp( - (x-a)*(x-a)/(b*b) );
    }
    double isSmallerOrEqual(double x) {
        if(x<a) return 1; return Math.exp( - (x-a)*(x-a)/(b*b) );
    }
    double center() {
        return a;
    }
}
```

```

double basis() {
    return b;
}
}

```

Sigma

```

private class MF_xfl_sigma extends InnerMembershipFunction {
    double a;
    double b;

    MF_xfl_sigma( double min, double max, double step, double param[]){
        super.min = min;
        super.max = max;
        super.step = step;
        this.a = param[0];
        this.b = param[1];
    }
    double param(int _i) {
        switch(_i) {
            case 0: return a;
            case 1: return b;
            default: return 0;
        }
    }
    double isEqual(double x) {
        return 1/(1+ Math.exp( (a-x)/b ));
    }
    double isGreaterOrEqual(double x) {
        double y = (b>0? x : min);
        return 1 / (1+ Math.exp( (a-y)/b ));
    }
    double isSmallerOrEqual(double x) {
        double y = (b<0? x : max);
        return 1 / (1+ Math.exp( (a-y)/b ));
    }
}

```

Método de defuzzificación media de los máximos:

```

public class xfl_defuz_MeanOfMaxima extends DefuzMethod {
    public xfl_defuz_MeanOfMaxima() {
        super.parameter = new Parameter[0];
        super.name = "MeanOfMaxima";
        super.pkg = "xfl";
    }

    public double compute(AggregateMemFunc mf) {
        double min = mf.min();
        double max = mf.max();
        double step = mf.step();
        double out = min, maximum = 0;
        int counter = 1;
        for(double x=min; x<=max; x+=step) {
            double m = mf.compute(x);

```



```

        if(m == maximum) { out+=x; counter++; }
        if(m > maximum) { maximum = m; out = x; counter = 1; }
    }
    return out/counter;
}
}

```

Método de defuzzificación máximo de los máximos:

```

public class xfl_defuz_LastOfMaxima extends DefuzMethod {
    public xfl_defuz_LastOfMaxima () {
        super.parameter = new Parameter[0];
        super.name = "LastOfMaxima";
        super.pkg = "xfl";
    }

    public double compute(AggregateMemFunc mf) {
        double min = mf.min();
        double max = mf.max();
        double step = mf.step();
        double out=min, maximum = 0;
        for(double x=min; x<=max; x+=step) {
            double m = mf.compute(x);
            if(m>=maximum) { maximum = m; out = x; }
        }
        return out;
    }

    public boolean test () {
        return true;
    }

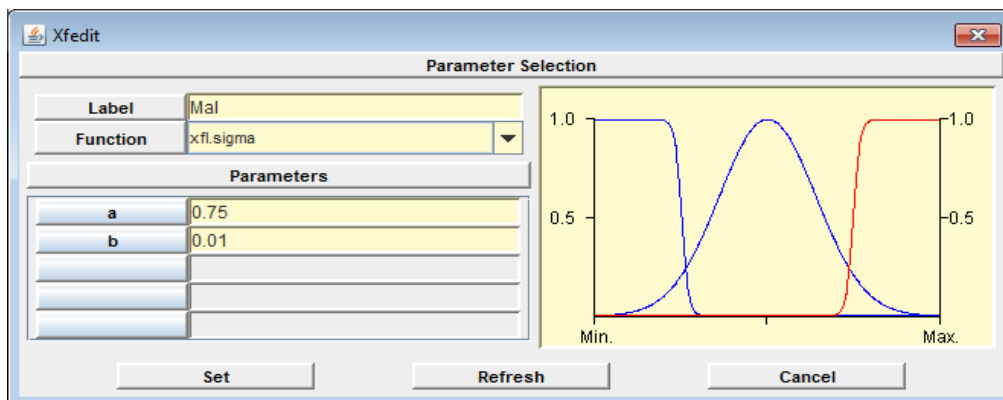
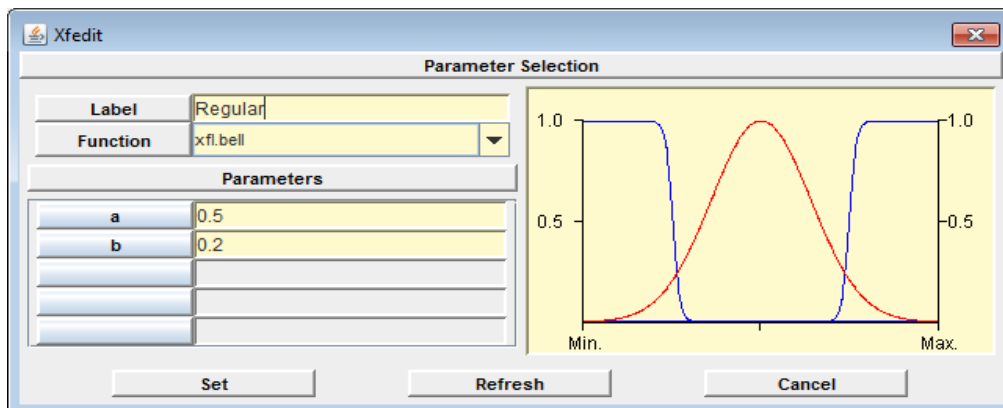
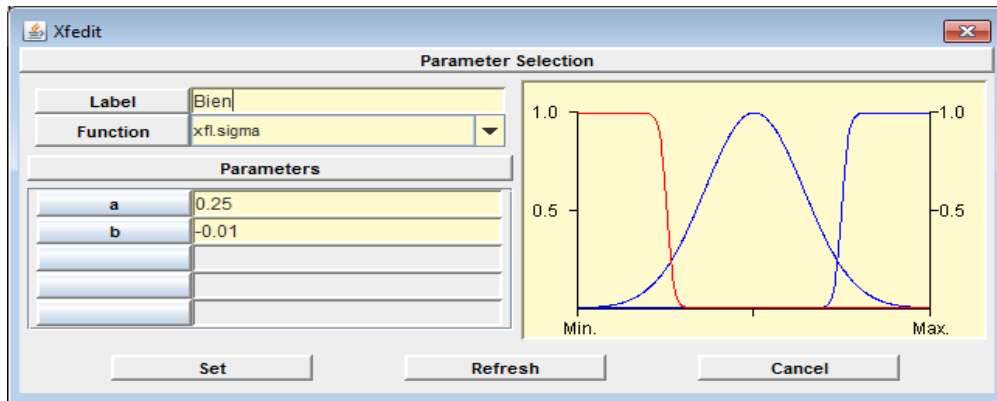
    public boolean test(AggregateMemFunc mf) {
        return true;
    }
}

```

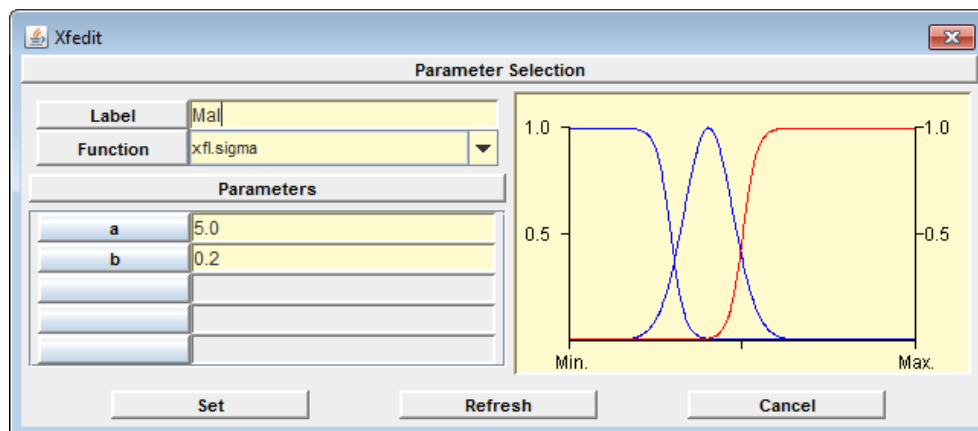
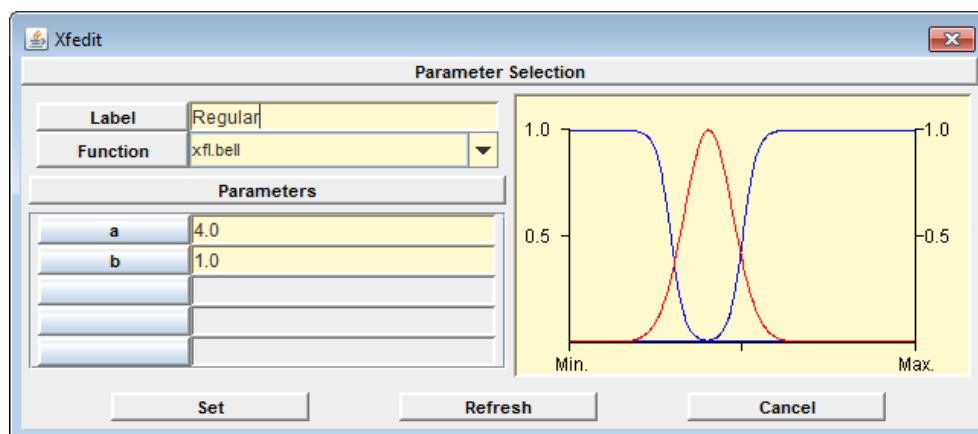
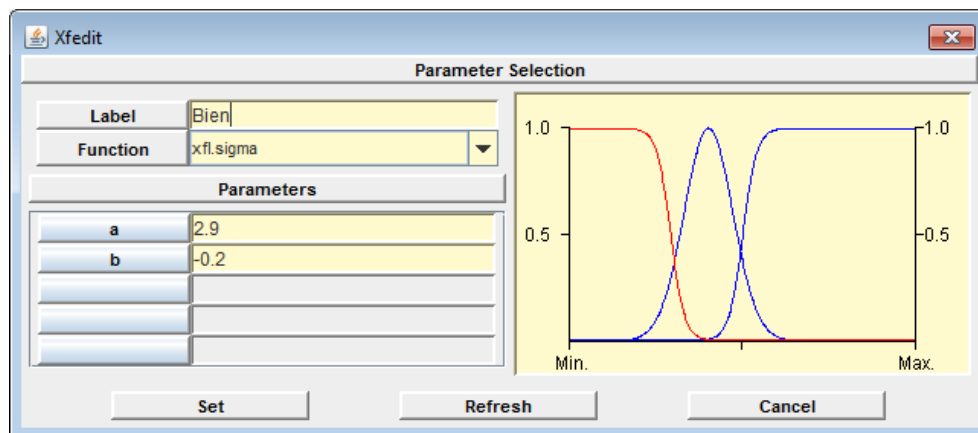
ANEXO II: Modelación borrosa en Xfuzzy 3.0.

Funciones de pertenencia:

- inSS:



- inSS, outOE, inRB, inRConst, outNivDH: similar a inSS.
- inSHU:



Reglas de inferencia de las distintas bases de reglas del SIB AgravamientoImpacto:

```

rulebase Preparacion (typeSistemAlertTemp inSAT, typeProteccMomentEmerg
inPME, typeComunEntrenProteg inCEP, typeComunicacion inComunic :
typePreparacion outPrep) {
    if(inSAT == Bien & inPME == Bien & inCEP == Bien & inComunic == Bien)
-> outPrep = Bien;
    if(inSAT == Mal & inPME == Mal & inCEP == Mal & inComunic == Mal) ->
outPrep = Mal;
    if(inSAT == Regular & inPME == Regular & inCEP == Regular & inComunic
== Regular) -> outPrep = Regular;
    if(inSAT == Bien & inPME == Bien & inCEP == Bien & inComunic ==
Regular) -> outPrep = Bien;
    if(inSAT == Bien & inPME == Bien & inCEP == Bien & inComunic == Mal)
-> outPrep = Regular;
    if(inSAT == Bien & inPME == Bien & inCEP == Regular & inComunic ==
Bien) -> outPrep = Bien;
    if(inSAT == Bien & inPME == Bien & inCEP == Mal & inComunic == Bien)
-> outPrep = Regular;
    if(inSAT == Bien & inPME == Regular & inCEP == Bien & inComunic ==
Bien) -> outPrep = Bien;
    if(inSAT == Bien & inPME == Mal & inCEP == Bien & inComunic == Bien)
-> outPrep = Regular;
    if(inSAT == Regular & inPME == Bien & inCEP == Bien & inComunic ==
Bien) -> outPrep = Bien;
    if(inSAT == Mal & inPME == Bien & inCEP == Bien & inComunic == Bien)
-> outPrep = Regular;
    if(inSAT == Bien & inPME == Bien & inCEP == Regular & inComunic ==
Regular) -> outPrep = Regular;
    if(inSAT == Bien & inPME == Bien & inCEP == Regular & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Bien & inCEP == Mal & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Bien & inCEP == Mal & inComunic == Mal) -
> outPrep = Mal;
    if(inSAT == Regular & inPME == Regular & inCEP == Bien & inComunic ==
Bien) -> outPrep = Regular;
    if(inSAT == Regular & inPME == Mal & inCEP == Bien & inComunic ==
Bien) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Regular & inCEP == Bien & inComunic ==
Bien) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Mal & inCEP == Bien & inComunic == Bien) -
> outPrep = Mal;
    if(inSAT == Bien & inPME == Regular & inCEP == Regular & inComunic ==
Bien) -> outPrep = Regular;
    if(inSAT == Bien & inPME == Regular & inCEP == Mal & inComunic ==
Bien) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Regular & inComunic ==
Bien) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Mal & inComunic == Bien) -
> outPrep = Mal;
    if(inSAT == Regular & inPME == Bien & inCEP == Bien & inComunic ==
Regular) -> outPrep = Regular;
    if(inSAT == Regular & inPME == Bien & inCEP == Bien & inComunic ==
Mal) -> outPrep = Mal;

```

```

    if(inSAT == Mal & inPME == Bien & inCEP == Bien & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Bien & inCEP == Bien & inComunic == Mal) -
> outPrep = Mal;
    if(inSAT == Bien & inPME == Regular & inCEP == Bien & inComunic ==
Regular) -> outPrep = Regular;
    if(inSAT == Bien & inPME == Regular & inCEP == Bien & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Bien & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Bien & inComunic == Mal) -
> outPrep = Mal;
    if(inSAT == Regular & inPME == Bien & inCEP == Regular & inComunic ==
Bien) -> outPrep = Regular;
    if(inSAT == Regular & inPME == Bien & inCEP == Mal & inComunic ==
Bien) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Bien & inCEP == Regular & inComunic ==
Bien) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Bien & inCEP == Mal & inComunic == Bien) -
> outPrep = Mal;
    if(inSAT == Bien & inPME == Regular & inCEP == Regular & inComunic ==
Regular) -> outPrep = Regular;
    if(inSAT == Bien & inPME == Regular & inCEP == Regular & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Regular & inCEP == Mal & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Regular & inCEP == Mal & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Regular & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Regular & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Mal & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Bien & inPME == Mal & inCEP == Mal & inComunic == Mal) ->
outPrep = Mal;
    if(inSAT == Regular & inPME == Bien & inCEP == Regular & inComunic ==
Regular) -> outPrep = Regular;
    if(inSAT == Regular & inPME == Bien & inCEP == Regular & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Regular & inPME == Bien & inCEP == Mal & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Regular & inPME == Bien & inCEP == Mal & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Bien & inCEP == Regular & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Bien & inCEP == Regular & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Bien & inCEP == Mal & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Bien & inCEP == Mal & inComunic == Mal) ->
outPrep = Mal;
    if(inSAT == Regular & inPME == Regular & inCEP == Bien & inComunic ==
Regular) -> outPrep = Regular;
    if(inSAT == Regular & inPME == Regular & inCEP == Bien & inComunic ==
Mal) -> outPrep = Mal;

```

```

    if(inSAT == Regular & inPME == Mal & inCEP == Bien & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Regular & inPME == Mal & inCEP == Bien & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Regular & inCEP == Bien & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Regular & inCEP == Bien & inComunic ==
Mal) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Mal & inCEP == Bien & inComunic ==
Regular) -> outPrep = Mal;
    if(inSAT == Mal & inPME == Mal & inCEP == Bien & inComunic == Mal) ->
outPrep = Mal;
    if(inSAT == Regular & inPME == Regular & inCEP == Regular & inComunic
== Bien) -> outPrep = Regular;
    if(inSAT == Mal & inPME == Mal & inCEP == Mal & inComunic == Bien) ->
outPrep = Mal;
}

rulebase Proteccion (typeBarreraNatural inBN, typeBarreraArtificial
inBA : typeProteccion outProtec) {
    if(inBN == Eficiente & inBA == Eficiente) -> outProtec = Bien;
    if(inBN == Eficiente & inBA == MedianamenteEficiente) -> outProtec =
Regular;
    if(inBN == MedianamenteEficiente & inBA == Eficiente) -> outProtec =
Bien;
    if(inBN == MedianamenteEficiente & inBA == MedianamenteEficiente) ->
outProtec = Regular;
    if(inBN == Eficiente & inBA == Deficiente) -> outProtec = Regular;
    if(inBN == Deficiente & inBA == Eficiente) -> outProtec = Regular;
    if(inBN == MedianamenteEficiente & inBA == Deficiente) -> outProtec =
Mal;
    if(inBN == Deficiente & inBA == MedianamenteEficiente) -> outProtec =
Mal;
    if(inBN == Deficiente & inBA == Deficiente) -> outProtec = Mal;
}

rulebase ReceptAmbientVulnerables (typeEspacNaturProteg inENP,
typeEspeciesProteg inEP, typePatrimHistArt inPHA :
typeReceptAmbientVulnerables outRAVul) {
    if(inENP == Bien & inEP == Bien & inPHA == Bien) -> outRAVul = Bien;
    if(inENP == Bien & inEP == Bien & inPHA == Regular) -> outRAVul =
Bien;
    if(inENP == Bien & inEP == Bien & inPHA == Mal) -> outRAVul =
Regular;
    if(inENP == Bien & inEP == Regular & inPHA == Bien) -> outRAVul =
Bien;
    if(inENP == Bien & inEP == Regular & inPHA == Regular) -> outRAVul =
Regular;
    if(inENP == Bien & inEP == Regular & inPHA == Mal) -> outRAVul = Mal;
    if(inENP == Bien & inEP == Mal & inPHA == Bien) -> outRAVul =
Regular;
    if(inENP == Bien & inEP == Mal & inPHA == Regular) -> outRAVul = Mal;
    if(inENP == Bien & inEP == Mal & inPHA == Mal) -> outRAVul = Mal;
    if(inENP == Regular & inEP == Bien & inPHA == Bien) -> outRAVul =
Bien;

```

```

    if(inENP == Regular & inEP == Bien & inPHA == Regular) -> outRAVul =
Regular;
    if(inENP == Regular & inEP == Bien & inPHA == Mal) -> outRAVul = Mal;
    if(inENP == Regular & inEP == Regular & inPHA == Bien) -> outRAVul =
Regular;
    if(inENP == Regular & inEP == Regular & inPHA == Regular) -> outRAVul
= Regular;
    if(inENP == Regular & inEP == Regular & inPHA == Mal) -> outRAVul =
Mal;
    if(inENP == Regular & inEP == Mal & inPHA == Bien) -> outRAVul = Mal;
    if(inENP == Regular & inEP == Mal & inPHA == Regular) -> outRAVul =
Mal;
    if(inENP == Regular & inEP == Mal & inPHA == Mal) -> outRAVul = Mal;
    if(inENP == Mal & inEP == Bien & inPHA == Bien) -> outRAVul =
Regular;
    if(inENP == Mal & inEP == Bien & inPHA == Regular) -> outRAVul = Mal;
    if(inENP == Mal & inEP == Bien & inPHA == Mal) -> outRAVul = Mal;
    if(inENP == Mal & inEP == Regular & inPHA == Bien) -> outRAVul = Mal;
    if(inENP == Mal & inEP == Regular & inPHA == Regular) -> outRAVul =
Mal;
    if(inENP == Mal & inEP == Regular & inPHA == Mal) -> outRAVul = Mal;
    if(inENP == Mal & inEP == Mal & inPHA == Bien) -> outRAVul = Mal;
    if(inENP == Mal & inEP == Mal & inPHA == Regular) -> outRAVul = Mal;
    if(inENP == Mal & inEP == Mal & inPHA == Mal) -> outRAVul = Mal;
}

```

```

rulebase Falta_de_Homeostasis (typePreparacion outPrep, typeProteccion
outProtec, typeReceptAmbientVulnerables outRAVul,
typeEstadoFisicoConstruc inEFC, typeDensidadPoblacion inDP,
typeAreaComunitaria inAC : typeFaltaHomeostasis outFaltaHomeostasis) {
    if((inDP == Alta & inAC == Mal & outPrep == Mal) | (outProtec == Mal
& outRAVul == Mal & inEFC == Mal)) -> outFaltaHomeostasis = Mal;
    if(inDP == Media & inAC == Regular & outPrep == Regular & outProtec
== Regular & outRAVul == Regular & inEFC == Regular) ->
outFaltaHomeostasis = Regular;
    if(outPrep == Mal & outProtec == Mal & outRAVul == Mal & inEFC ==
Mal) -> outFaltaHomeostasis = Mal;
    if(inAC == Mal & outPrep == Mal & outProtec == Mal & outRAVul == Mal
& inEFC == Mal) -> outFaltaHomeostasis = Mal;
    if(inDP == Media & inAC == Regular & outPrep == Regular & outProtec
== Mal & outRAVul == Mal & inEFC == Mal) -> outFaltaHomeostasis =
Regular;
    if(inDP == Media & inAC == Mal & outPrep == Mal & outProtec ==
Regular & outRAVul == Regular & inEFC == Regular) ->
outFaltaHomeostasis = Regular;
    if(inDP == Alta & inAC == Regular & outPrep == Regular & outProtec ==
Regular & outRAVul == Regular & inEFC == Regular) ->
outFaltaHomeostasis = Regular;
    if(inDP == Alta & inAC == Bien & outPrep == Bien & outProtec ==
Regular & outRAVul == Regular & inEFC == Regular) ->
outFaltaHomeostasis = Regular;
    if(inDP == Alta & inAC == Regular & outPrep == Regular & outProtec ==
Bien & outRAVul == Bien & inEFC == Bien) -> outFaltaHomeostasis =
Regular;
    if(inDP == Alta & inAC == Bien & outPrep == Bien & outProtec == Bien
& outRAVul == Bien & inEFC == Bien) -> outFaltaHomeostasis = Regular;
}

```

```

    if(inDP == Media & inAC == Bien & outPrep == Bien & outProtec == Bien
    & outRAVul == Bien & inEFC == Bien) -> outFaltaHomeostasis = Bien;
    if(inDP == Baja & inAC == Bien & outPrep == Bien & outProtec == Bien
    & outRAVul == Bien & inEFC == Bien) -> outFaltaHomeostasis = Bien;
    if(inDP == Baja & inAC == Regular & outPrep == Regular & outProtec ==
    Bien & outRAVul == Bien & inEFC == Bien) -> outFaltaHomeostasis = Bien;
    if(inDP == Baja & inAC == Bien & outPrep == Bien & outProtec ==
    Regular & outRAVul == Regular & inEFC == Regular) ->
    outFaltaHomeostasis = Bien;
}

```

```

rulebase OperatividadEmergencias (typeBrigLuchaIncendios inBLI,
typeBomberos inBomb, typeAmbulancias inAmbul, typePrimAuxyOtros inPAYO
: typeOperatEmerg outOE) {
    if(inBLI == MaxOperat & inBomb == MaxOperat & inAmbul == MaxOperat &
    inPAYO == MaxOperat) -> outOE = Bien;
    if(inBLI == MaxOperat & inBomb == MaxOperat & inAmbul == MaxOperat &
    inPAYO == MediaOperat) -> outOE = Bien;
    if(inBLI == MaxOperat & inBomb == MaxOperat & inAmbul == MediaOperat
    & inPAYO == MaxOperat) -> outOE = Bien;
    if(inBLI == MaxOperat & inBomb == MediaOperat & inAmbul == MaxOperat
    & inPAYO == MaxOperat) -> outOE = Bien;
    if(inBLI == MediaOperat & inBomb == MaxOperat & inAmbul == MaxOperat
    & inPAYO == MaxOperat) -> outOE = Bien;
    if(inBLI == MaxOperat & inBomb == MaxOperat & inAmbul == MediaOperat
    & inPAYO == MediaOperat) -> outOE = Regular;
    if(inBLI == MaxOperat & inBomb == MediaOperat & inAmbul ==
    MediaOperat & inPAYO == MediaOperat) -> outOE = Regular;
    if(inBLI == MediaOperat & inBomb == MediaOperat & inAmbul ==
    MediaOperat & inPAYO == MediaOperat) -> outOE = Regular;
    if(inBLI == MediaOperat & inBomb == MediaOperat & inAmbul ==
    MediaOperat & inPAYO == BajaOperat) -> outOE = Regular;
    if(inBLI == MediaOperat & inBomb == MediaOperat & inAmbul ==
    BajaOperat & inPAYO == MediaOperat) -> outOE = Regular;
    if(inBLI == MediaOperat & inBomb == BajaOperat & inAmbul ==
    MediaOperat & inPAYO == MediaOperat) -> outOE = Regular;
    if(inBLI == BajaOperat & inBomb == MediaOperat & inAmbul ==
    MediaOperat & inPAYO == MediaOperat) -> outOE = Regular;
    if(inBLI == MediaOperat & inBomb == MediaOperat & inAmbul ==
    BajaOperat & inPAYO == BajaOperat) -> outOE = Mal;
    if(inBLI == MediaOperat & inBomb == BajaOperat & inAmbul ==
    BajaOperat & inPAYO == MediaOperat) -> outOE = Mal;
    if(inBLI == BajaOperat & inBomb == BajaOperat & inAmbul ==
    MediaOperat & inPAYO == MediaOperat) -> outOE = Mal;
    if(inBLI == BajaOperat & inBomb == MediaOperat & inAmbul ==
    MediaOperat & inPAYO == BajaOperat) -> outOE = Mal;
    if(inBLI == BajaOperat & inBomb == BajaOperat & inAmbul == BajaOperat
    & inPAYO == BajaOperat) -> outOE = Mal;
}

```

```

rulebase NivelDesarrolloHumano (typeLongevidad inLong, typeEscolaridad
inEscol, typeIngresoPerCap inIPC : typeNivelDesarrHum outNivDH) {
    if(inLong == Baja & inEscol == Alta & inIPC == Bien) -> outNivDH =
    Alto;
    if(inLong == Baja & inEscol == Media & inIPC == Regular) -> outNivDH
    = Medio;
}

```



```

    if(inLong == Media & inEscol == Media & inIPC == Regular) -> outNivDH
= Medio;
    if(inLong == Media & inEscol == Alta & inIPC == Bien) -> outNivDH =
Alto;
    if(inLong == Alta & inEscol == Media & inIPC == Regular) -> outNivDH
= Medio;
    if(inLong == Alta & inEscol == Baja & inIPC == Mal) -> outNivDH =
Bajo;
    if(inLong == Alta & inEscol == Baja & inIPC == Regular) -> outNivDH =
Bajo;
    if(inLong == Media & inEscol == Baja & inIPC == Mal) -> outNivDH =
Bajo;
}

```

```

rulebase Falta_de_Resiliencia (typeServiciosSalv inSS, typeOperatEmerg
outOE, typeServHospUrgenc inSHU, typeRecuperBasica inRB,
typeNivelDesarrHum outNivDH, typeReconstruccion inRConst :
typeFaltaResiliencia outFaltaResiliencia) {
    if((inSS == Mal & outOE == Mal) | (inSS == Mal & inSHU == Mal) |
(inSHU == Mal & outOE == Mal)) -> outFaltaResiliencia = Mal;
    if((inSS == Regular | outOE == Regular | inSHU == Regular) & (inRB ==
Mal | outNivDH == Bajo | inRConst == Mal)) -> outFaltaResiliencia =
Mal;
    if((inSS == Bien | outOE == Bien | inSHU == Bien) & inRB == Mal &
outNivDH == Bajo & inRConst == Mal) -> outFaltaResiliencia = Regular;
    if((inSS == Bien | outOE == Bien | inSHU == Bien) & (inRB == Mal |
outNivDH == Bajo | inRConst == Mal)) -> outFaltaResiliencia = Bien;
    if(inSS == Regular & outOE == Regular & inSHU == Regular & inRB ==
Regular & outNivDH == Medio & inRConst == Regular) ->
outFaltaResiliencia = Regular;
    if(inSS == Bien | outOE == Bien | inSHU == Bien) & (inRB == Regular
| outNivDH == Medio | inRConst == Regular)) -> outFaltaResiliencia =
Bien;
    if(inSS == Bien & outOE == Bien & inSHU == Bien & inRB == Bien &
outNivDH == Alto & inRConst == Bien) -> outFaltaResiliencia = Bien;
}

```

Reglas de inferencia del SIB VulnerabilidadExplosion:

```

rulebase Rule_Vulner (typeGravIndiceFallec inGravIndiceFallec,
typeGravIndiceLesion inGravIndiceLesion, typeDInterMayores
inDInterMayores, typeDExterMayores inDExterMayores, typeDMedioambiental
inDMedioambiental : typeVulnerabilidad outVulnerabilidad) {
    if(inGravIndiceFallec == Alto) -> outVulnerabilidad = MuyAlta;
    if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Bajo &
inDInterMayores == Bajo & inDExterMayores == Bajo & inDMedioambiental
== Bajo) -> outVulnerabilidad = MuyBaja;
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio &
inDInterMayores == Alto & inDExterMayores == Alto & inDMedioambiental
== Alto) -> outVulnerabilidad = MuyAlta;
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Alto) ->
outVulnerabilidad = MuyAlta;
}

```

```

    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio &
inDInterMayores == Medio & inDExterMayores == Alto & inDMedioambiental
== Alto) -> outVulnerabilidad = Alta;
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio &
inDInterMayores == Medio & inDExterMayores == Medio & inDMedioambiental
== Medio) -> outVulnerabilidad = Media;
    if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Bajo &
inDInterMayores == Medio) -> outVulnerabilidad = Baja;
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Bajo &
inDInterMayores == Bajo & inDExterMayores == Bajo & inDMedioambiental
== Bajo) -> outVulnerabilidad = Media;
}

```

Reglas de inferencia del SIB VulnerabilidadDerrame:

```

rulebase Rule_Vulner (typeGravIndiceFallec inGravIndiceFallec,
typeGravIndiceLesion inGravIndiceLesion, typeDInterMayores
inDInterMayores, typeDExterMayores inDExterMayores, typeDMedioambiental
inDMedioambiental : typeVulnerabilidad outVulnerabilidad) {
    if(inGravIndiceFallec == Alto & inGravIndiceLesion == Alto) ->
outVulnerabilidad = MuyAlta;
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio &
inDInterMayores == Alto & inDExterMayores == Alto & inDMedioambiental
== Alto) -> outVulnerabilidad = MuyAlta;
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Medio &
inDMedioambiental == Medio) -> outVulnerabilidad = Media;
    if(inGravIndiceFallec == Alto & inGravIndiceLesion == Medio &
inDMedioambiental == Medio) -> outVulnerabilidad = Alta;
    if(inGravIndiceFallec == Alto & inGravIndiceLesion == Medio &
inDMedioambiental == Bajo) -> outVulnerabilidad = Media;
    if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Alto &
inDMedioambiental == Medio) -> outVulnerabilidad = Media;
    if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Bajo &
inDInterMayores == Alto & inDExterMayores == Alto & inDMedioambiental
== Bajo) -> outVulnerabilidad = Baja;
    if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Bajo &
inDInterMayores == Medio & inDExterMayores == Medio & inDMedioambiental
== Bajo) -> outVulnerabilidad = Baja;
    if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Bajo &
inDInterMayores == Bajo & inDExterMayores == Bajo & inDMedioambiental
== Bajo) -> outVulnerabilidad = MuyBaja;
}

```

Reglas de inferencia del SIB VulnerabilidadEscape:

```

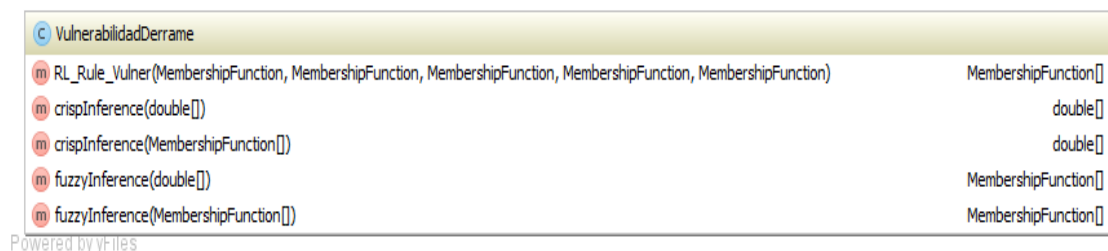
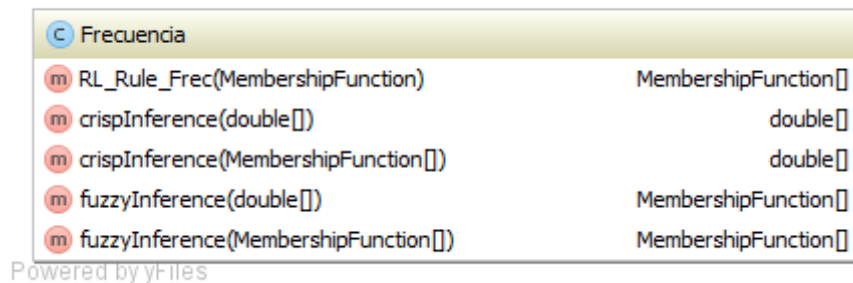
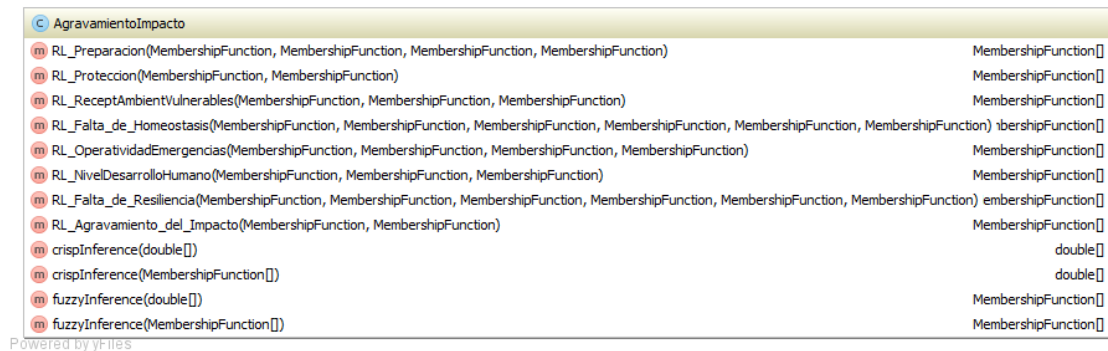
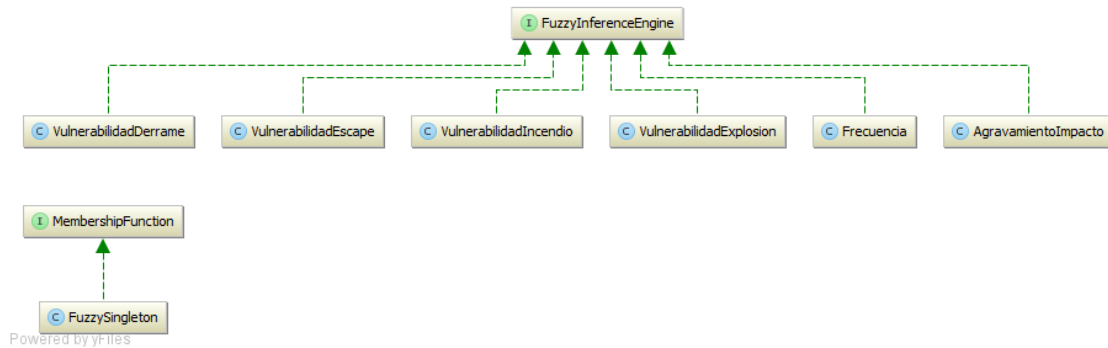
rulebase Rule_Vulner (typeGravIndiceFallec inGravIndiceFallec,
typeGravIndiceLesion inGravIndiceLesion, typeDInterMayores
inDInterMayores, typeDExterMayores inDExterMayores, typeDMedioambiental
inDMedioambiental : typeVulnerabilidad outVulnerabilidad) {
    if(inGravIndiceFallec == Alto & inGravIndiceLesion == Alto) ->
outVulnerabilidad = MuyAlta;
    if(inGravIndiceFallec == Alto & inDInterMayores == Alto &
inDExterMayores == Alto & inDMedioambiental == Alto) ->
outVulnerabilidad = MuyAlta;
}

```

```
    if(inGravIndiceFallec == Alto & inGravIndiceLesion == Medio &  
inDMedioambiental == Medio) -> outVulnerabilidad = Alta;  
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Alto &  
inDMedioambiental == Alto) -> outVulnerabilidad = Alta;  
    if(inGravIndiceFallec == Bajo & inGravIndiceLesion == Alto &  
inDMedioambiental == Alto) -> outVulnerabilidad = Media;  
    if(inGravIndiceFallec == Medio & inGravIndiceLesion == Alto &  
inDMedioambiental == Medio) -> outVulnerabilidad = Media;  
}
```

ANEXO III: Diagramas de clases de SIDCIRT

Es necesario poner las clases con su información por separado debido a su tamaño.



C VulnerabilidadEscape		
m	RL_Rule_Vulner(MembershipFunction, MembershipFunction, MembershipFunction, MembershipFunction, MembershipFunction)	MembershipFunction[]
m	crispInference(double[])	double[]
m	crispInference(MembershipFunction[])	double[]
m	fuzzyInference(double[])	MembershipFunction[]
m	fuzzyInference(MembershipFunction[])	MembershipFunction[]

Powered by yFiles

C VulnerabilidadExplosion		
m	RL_Rule_Vulner(MembershipFunction, MembershipFunction, MembershipFunction, MembershipFunction, MembershipFunction)	MembershipFunction[]
m	crispInference(double[])	double[]
m	crispInference(MembershipFunction[])	double[]
m	fuzzyInference(double[])	MembershipFunction[]
m	fuzzyInference(MembershipFunction[])	MembershipFunction[]

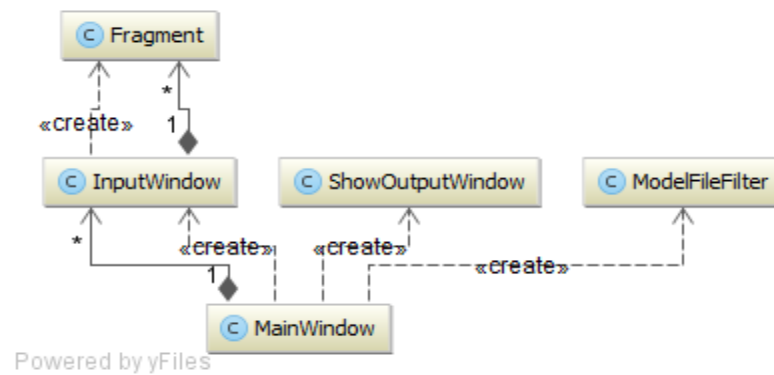
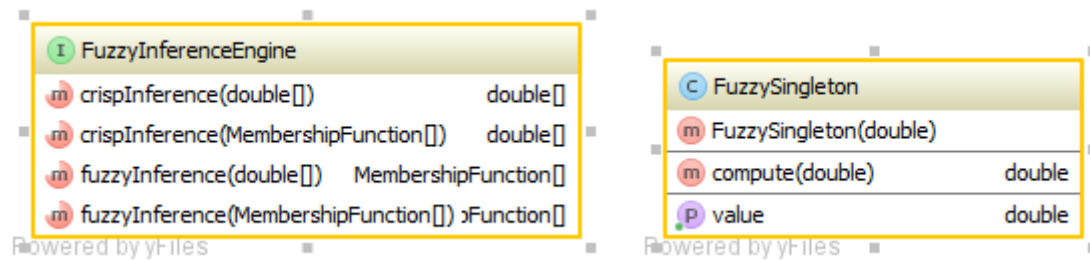
Powered by yFiles

C VulnerabilidadIncendio		
m	RL_Rule_Vulner(MembershipFunction, MembershipFunction, MembershipFunction, MembershipFunction, MembershipFunction)	MembershipFunction[]
m	crispInference(double[])	double[]
m	crispInference(MembershipFunction[])	double[]
m	fuzzyInference(double[])	MembershipFunction[]
m	fuzzyInference(MembershipFunction[])	MembershipFunction[]

Powered by yFiles

C Fragment		
f	min	double
f	max	double
f	converter	Converter<Integer, String>
f	bindingGroup	BindingGroup
m	Fragment(double, double)	
m	stringToInt(String)	int
m	intToString(int)	String
m	initComponents()	void
P	label	JLabel
P	slider	JSlider
P	textField	JTextField

Powered by yFiles

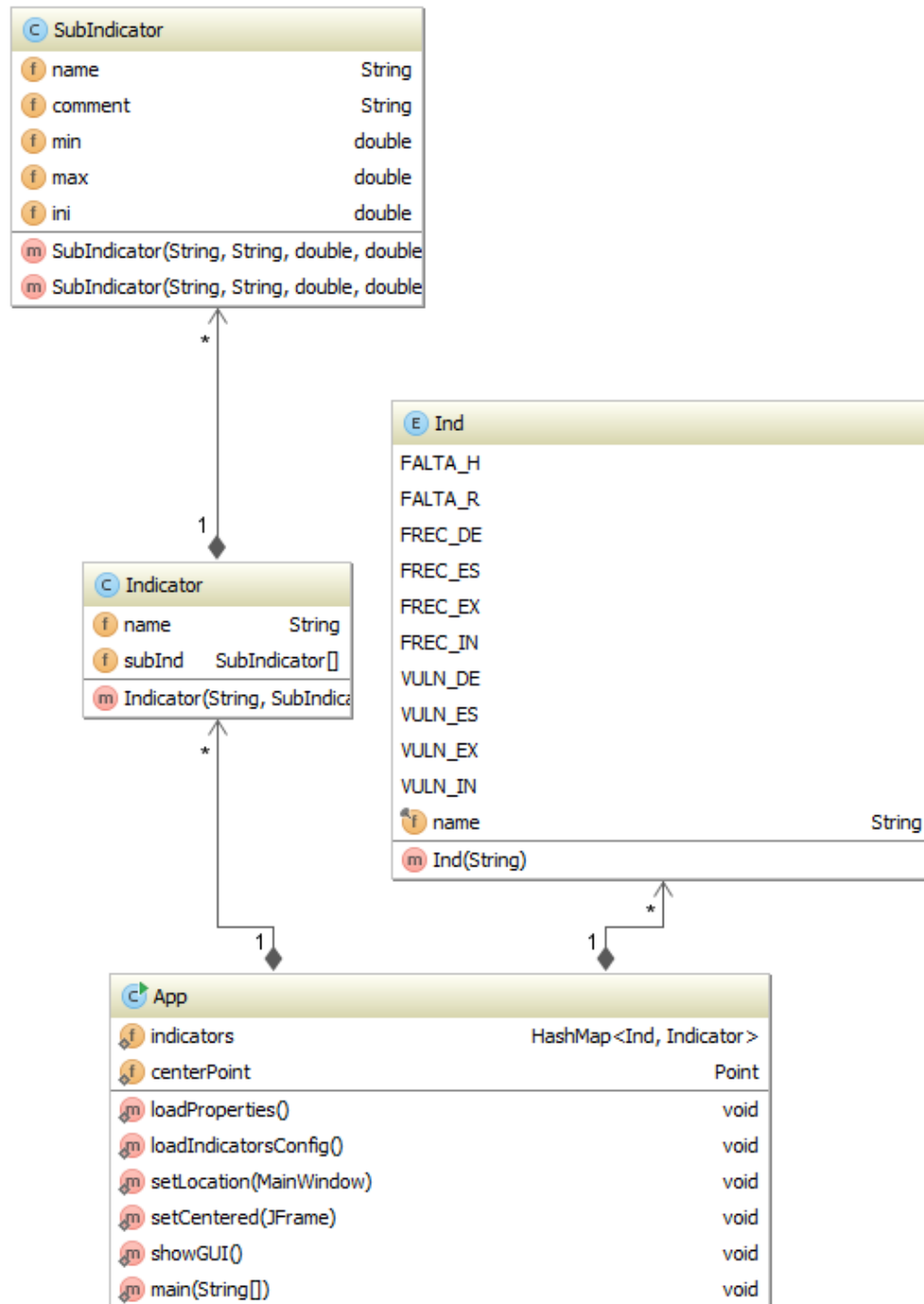


C MainWindow	
i indWindow	HashMap<Ind, InputWindow>
i appIcon	ImageIcon
i chooser	JFileChooser
i E	int
i cai	double
i frec	double[]
i vuln	double[]
i aboutMenuItem	JMenuItem
i caiMenu	JMenu
i calculateMenu	JMenu
i closeMenuItem	JMenuItem
i editMenu	JMenu
i falta_hMenuItem	JMenuItem
i falta_rMenuItem	JMenuItem
i fileMenu	JMenu
i frecMenu	JMenu
i frec_deMenuItem	JMenuItem
i frec_esMenuItem	JMenuItem
i frec_exMenuItem	JMenuItem
i frec_inMenuItem	JMenuItem
i helpMenu	JMenu
i helpMenuItem	JMenuItem
i irtMenuItem	JMenuItem
i jSeparator1	Separator
i jSeparator2	Separator
i menuBar	JMenuBar
i openModelMenuItem	JMenuItem
i saveModelMenuItem	JMenuItem
i toolBar	JToolBar
i vulnMenu	JMenu
i vuln_deMenuItem	JMenuItem
i vuln_esMenuItem	JMenuItem
i vuln_exMenuItem	JMenuItem
i vuln_inMenuItem	JMenuItem
m MainWindow()	
m initInputWindows()	void
m infer(FuzzyInferenceEngine, double[])	double[]
m calculateCAI()	void
m calculateFrec()	void
m calculateVuln()	void
m calculateIRT()	double
m initComponents()	void
m openModelMenuItemActionPerformed(ActionEvent)	void
m helpMenuItemActionPerformed(ActionEvent)	void
m irtMenuItemActionPerformed(ActionEvent)	void
m closeMenuItemActionPerformed(ActionEvent)	void
m showIndicatorActionPerformed(ActionEvent)	void
m aboutMenuItemActionPerformed(ActionEvent)	void
m saveModelMenuItemActionPerformed(ActionEvent)	void

Powered by yFiles

I MembershipFunction
m compute(double) double

Powered by yFiles



Powered by yFiles

ShowOutputWindow		
class		String[]
message		String[]
clasTextField		JTextField
jButton1		JButton
jLabel1		JLabel
jLabel2		JLabel
jLabel3		JLabel
jScrollPane1		JScrollPane
messageTextArea		JTextArea
<hr/>		
ShowOutputWindow()		
showOutput()		void
initComponents()		void
jButton1ActionPerformed(ActionEvent)		void
outputTextFieldActionPerformed(ActionEvent)		void
<hr/>		
outputTextField		JTextField
irt		double

Powered by yFiles

ANEXO IV: Casos de estudio

Caso 1:

Falta de homeostasis

Sistema de Alerta Temprano	0.5	Protec Momento de Emerg	0.5
Comunidad Entrenada	0.5	Comunicación	0.5
Barrera Natural	2.5	Barrera Artificial	2.5
Espacios Naturales Proteg	0.5	Especies Protegidas	0.5
Patrimonio Histórico Artístico	0.5	Estado Físico de las Constr	0.5
Densidad de población	250.0	Área Comunitaria	0.5

Aceptar Cancelar

Falta de resiliencia

Servicios de salvamento	0.5	Brig de Lucha incendios	30.0
Bomberos	5.0	Ambulancias	5.0
Primeros Aux y Otros	30.0	Servicios Hosp Urgencias	5.0
Recuperación Básica	0.5	Longevidad	50.0
Escolaridad	0.5	Ingreso per Cápita	25.0
REconstrucción	0.5		

Aceptar Cancelar

Frecuencia de escape

Tiempo

Aceptar Cancelar

Frecuencia de derrame

Tiempo

Aceptar Cancelar

Frecuencia de explosión

Tiempo

Aceptar Cancelar

Frecuencia de incendio

Tiempo

Aceptar Cancelar

Vulnerabilidad de escape

Gravedad Indice de Fallecidos	<input type="text" value="0.5"/>	Gravedad Indice de Lesionados	<input type="text" value="0.5"/>
	<input type="range"/>		<input type="range"/>
Daños Externos Mayores	<input type="text" value="0.5"/>	Daños Internos Mayores	<input type="text" value="0.5"/>
	<input type="range"/>		<input type="range"/>
Daños Medioambientales	<input type="text" value="0.5"/>		
	<input type="range"/>		

Aceptar Cancelar

Vulnerabilidad de escape

Gravedad Índice de Fallecidos	0.5	Gravedad Índice de Lesionados	0.5
Daños Externos Mayores	0.5	Daños Internos Mayores	0.5
Daños Medioambientales	0.5		

Aceptar Cancelar

Vulnerabilidad de explosión

Gravedad Índice de Fallecidos	0.5	Gravedad Índice de Lesionados	0.5
Daños Externos Mayores	0.5	Daños Internos Mayores	0.5
Daños Medioambientales	0.5		

Aceptar Cancelar

Vulnerabilidad de incendio

Gravedad Índice de Fallecidos	0.5	Gravedad Índice de Lesionados	0.5
Daños Externos Mayores	0.5	Daños Internos Mayores	0.5
Daños Medioambientales	0.5		

Aceptar Cancelar

Caso 2:

Falta de homeostasis

Sistema de Alerta Temprano	0.87	Protec Momento de Emerg	0.87
Comunidad Entrenada	0.92	Comunicación	0.82
Barrera Natural	0.4	Barrera Artificial	0.5
Espacios Naturales Proteg	1.0	Especies Protegidas	0.86
Patrimonio Histórico Artístico	0.9	Estado Físico de las Constr	0.95
Densidad de población	395.0	Área Comunitaria	0.97

Aceptar Cancelar

Falta de resiliencia

Servicios de salvamento	0.93	Brig de Lucha incendios	50.4
Bomberos	8.4	Ambulancias	8.5
Primeros Aux y Otros	52.2	Servicios Hosp Urgencias	8.5
Recuperación Básica	0.89	Longevidad	87.0
Escolaridad	0.11	Ingreso per Cápita	11.5
REconstrucción	0.92		

Aceptar Cancelar

Frecuencia de escape

Tiempo

Aceptar Cancelar

Frecuencia de derrame

Tiempo

Aceptar Cancelar

Frecuencia de explosión

Tiempo

Aceptar Cancelar

Frecuencia de incendio

Tiempo

Aceptar Cancelar

Vulnerabilidad de escape

Gravedad Índice de Fallecidos	<input type="text" value="0.92"/>	Gravedad Índice de Lesionados	<input type="text" value="0.89"/>
	<input type="range"/>		<input type="range"/>
Daños Externos Mayores	<input type="text" value="1.0"/>	Daños Internos Mayores	<input type="text" value="0.92"/>
	<input type="range"/>		<input type="range"/>
Daños Medioambientales	<input type="text" value="0.95"/>		
	<input type="range"/>		

Aceptar Cancelar

Vulnerabilidad de derrame

Gravedad Índice de Fallecidos	0.91	Gravedad Índice de Lesionados	0.98
Daños Externos Mayores	0.96	Daños Internos Mayores	0.91
Daños Medioambientales	0.87		

Aceptar Cancelar

Vulnerabilidad de explosión

Gravedad Índice de Fallecidos	1.0	Gravedad Índice de Lesionados	0.85
Daños Externos Mayores	0.87	Daños Internos Mayores	0.97
Daños Medioambientales	1.0		

Aceptar Cancelar

Vulnerabilidad de incendio

Gravedad Índice de Fallecidos	1.0	Gravedad Índice de Lesionados	0.89
Daños Externos Mayores	0.89	Daños Internos Mayores	0.88
Daños Medioambientales	0.93		

Aceptar Cancelar

Caso 3:

Falta de homeostasis

Sistema de Alerta Temprano	0.17	Protec Momento de Emerg	0.02
Comunidad Entrenada	0.07	Comunicación	0.11
Barrera Natural	4.55	Barrera Artificial	4.5
Espacios Naturales Proteg	0.08	Especies Protegidas	0.1
Patrimonio Histórico Artístico	0.05	Estado Físico de las Constr	0.14
Densidad de población	60.0	Área Comunitaria	0.17

Falta de resiliencia

Servicios de salvamento	0.05	Brig de Lucha incendios	4.2
Bomberos	1.3	Ambulancias	0.5
Primeros Aux y Otros	1.2	Servicios Hosp Urgencias	0.3
Recuperación Básica	0.01	Longevidad	2.0
Escolaridad	0.89	Ingreso per Cápita	46.0
REconstrucción	0.07		

Frecuencia de escape

Tiempo

Frecuencia de derrame

Tiempo

Frecuencia de explosión

Tiempo

Frecuencia de incendio

Tiempo

Vulnerabilidad de escape

Gravedad Índice de Fallecidos	<input type="text" value="0.1"/>	Gravedad Índice de Lesionados	<input type="text" value="0.12"/>
	<input type="range" value="0.1"/>		<input type="range" value="0.12"/>
Daños Externos Mayores	<input type="text" value="0.02"/>	Daños Internos Mayores	<input type="text" value="0.04"/>
	<input type="range" value="0.02"/>		<input type="range" value="0.04"/>
Daños Medioambientales	<input type="text" value="0.12"/>		
	<input type="range" value="0.12"/>		

Vulnerabilidad de derrame

Gravedad Indice de Fallecidos	0.04	Gravedad Indice de Lesionados	0.02
Daños Externos Mayores	0.04	Daños Internos Mayores	0.02
Daños Medioambientales	0.08		

Aceptar Cancelar

Vulnerabilidad de explosión

Gravedad Indice de Fallecidos	0.1	Gravedad Indice de Lesionados	0.04
Daños Externos Mayores	0.08	Daños Internos Mayores	0.07
Daños Medioambientales	0.04		

Aceptar Cancelar

Vulnerabilidad de incendio

Gravedad Indice de Fallecidos	0.04	Gravedad Indice de Lesionados	0.08
Daños Externos Mayores	0.03	Daños Internos Mayores	0.08
Daños Medioambientales	0.05		

Aceptar Cancelar