

Universidad de las Ciencias Informáticas

Facultad 6



Trabajo de Diploma para optar por el título de Ingeniero en Ciencias
Informáticas

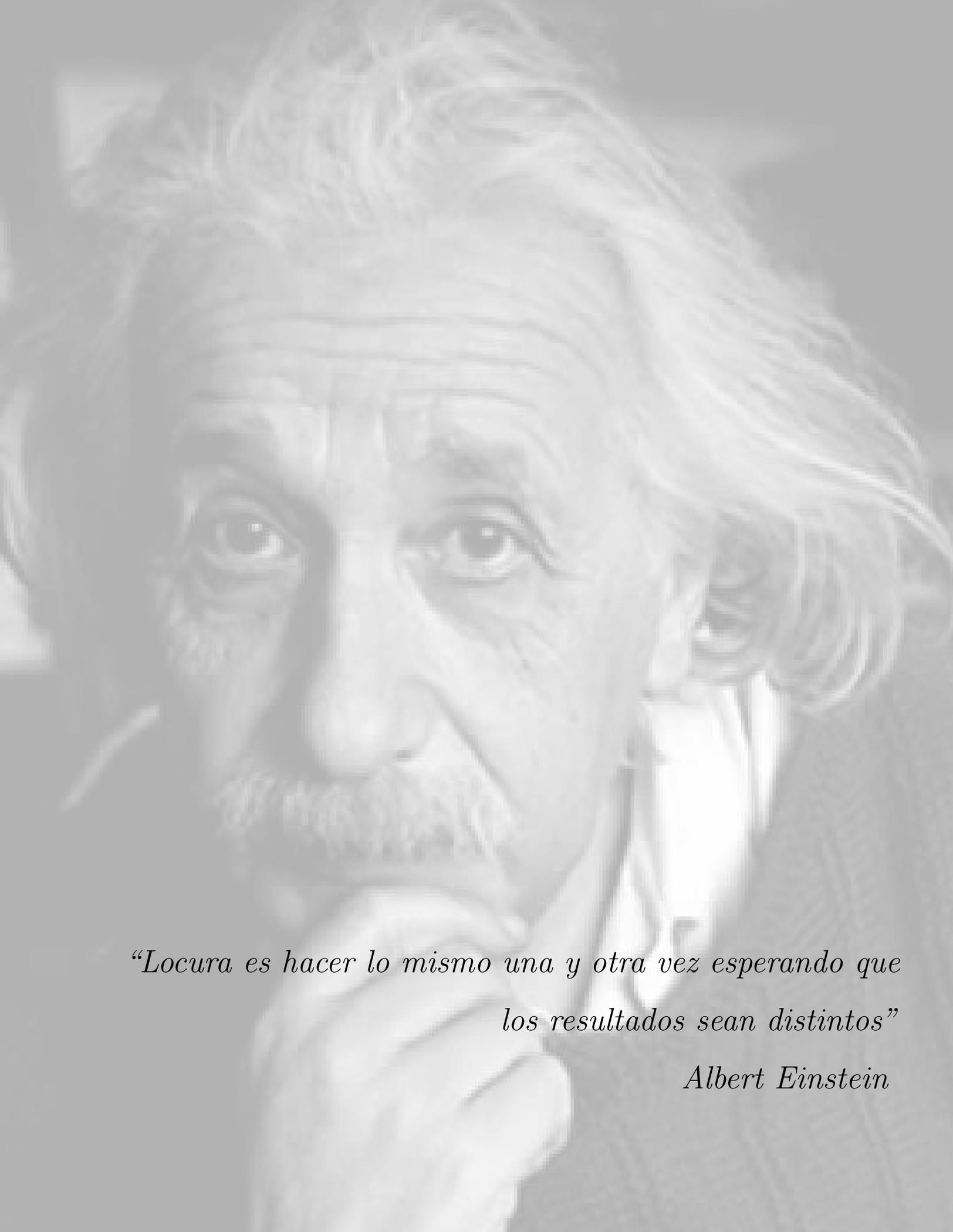
“Sistema de apoyo a la toma de decisiones basado en el modelo
computacional 2-tupla lingüística”

Autores: Dayron Mellado Martínez
Yoel Rojo Corrada

Tutores: Dra.C. Yeleny Zulueta Véliz
Ing. Willian Simón Grass

La Habana, junio de 2016

“Año 58 de la Revolución”



“Locura es hacer lo mismo una y otra vez esperando que los resultados sean distintos”

Albert Einstein

DECLARACIÓN DE AUTORÍA

Nosotros Dayron Mellado Martínez y Yoel Rojo Corrada declaramos ser los legítimos autores del trabajo titulado: “Sistema de Apoyo a la Toma de Decisiones basado en el modelo computacional 2 – tupla lingüística”, y reconocemos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año _____.

Firma del Autor
Dayron Mellado Martínez

Firma del Autor
Yoel Rojo Corrada

Firma de la Tutora
Dra.C. Yeleny Zulueta Véliz

Firma del Tutor
Ing. Willian Simón Grass

Síntesis de la Tutora:

Nombre y Apellidos: Dra.C. Yeleny Zulueta Véliz

Correo electrónico: yeleny@uci.cu

Año de graduado:

Profesión:

Breve descripción:

Síntesis del Tutor:

Nombre y Apellidos: Ing. Willian Simón Grass

Correo electrónico: wgrass@uci.cu.

Año de graduado:

Profesión: Ingeniero en Ciencias Informáticas.

Breve descripción:

Síntesis del Autor

Nombre y Apellidos: Dayron Mellado Martínez

Correo electrónico: dmulado@estudiantes.uci.cu

Síntesis del Autor

Nombre y Apellidos: Yoel Rojo Corrada

Correo electrónico: yrojo@estudiantes.uci.cu

Agradecimientos

Dayron

Con estas líneas quiero agradecer a todas las personas que de una forma u otra han contribuido en mi formación como futuro profesional, a mi familia especialmente a mi madre por apoyarme en cada momento de mi vida. A mi novia que ha apoyado en todo, para que la tesis saliera con la calidad requerida.

A los amigos que vienen desde mucho antes de la universidad Ivelisse, Karla, Pita, Alberto.

A los amigos de la Universidad especialmente a Yaisel, Darian, Eddy, Daynellis, Wendy y mi compañero de tesis.

A los profesores que he tenido en estos 5 años, al tribunal, a los tutores y en especial a la profe Aliana por hacer que algunas tareas se volvieran más sencillas.

Yoel

Aunque ahora todo lo pasado parezca fácil realmente no lo ha sido. Sin embargo, siempre hay personas dispuestas a hacer que los caminos más intransitables se hagan más sencillos de recorrer.

En primer lugar, se encuentran mis padres, los que ha sido mi fuente de inspiración para estar aquí hoy, a ustedes muchas gracias por su apoyo incondicional, por sus consejos y por estar siempre a mi lado. También agradecer a mi hermano que, a pesar de nuestros roces, me apoya muchísimo.

A mi novia Rachel muchas gracias por su ayuda y apoyo, por esas madrugadas en vela para que todo saliera lo mejor posible.

Agradecer también a mi compañero de tesis Dayron ya que sin él este logro hubiera sido mucho más difícil de alcanzar.

A mis suegros Mercedes y Rafael por acogerme con un hijo más y apoyarme en todo momento muchas gracias. A los tutores muchas gracias por la guía.

Muchas gracias a todos aquellos que de una forma u otra aportaron su granito de arena para que yo esté aquí hoy.

Dedicatoria

*Dedicado a mi familia especialmente a mi madre por su apoyo y dedicación,
a mi tía Nena y mi primo Tony por estar siempre presente a pesar de las distancias.*

Dayron

A mis padres, por guiarme siempre por el mejor de los caminos.

Yoel

Resumen

El modelo lingüístico 2-tupla es ampliamente utilizado en la resolución de problemas de Toma de Decisión Lingüística (TDL) debido a que ofrece resultados cercanos al lenguaje humano y provee un alto nivel de interpretabilidad, a la vez que permite implementar procesos de Computación con Palabras sin pérdida de información. Con el objetivo de contribuir a la resolución de problemas de TDL basados en el modelo 2-tupla lingüística, se desarrolló el sistema Web XABAL 2 TUPLE SOLVE. El sistema permite resolver problemas de TDL mediante un esquema de resolución basado en el análisis de decisión, que incluye tres fases principales: Definición del Marco del Problema, Recopilación de Información y Agregación de Información. En la primera fase se definen los elementos del problema: las alternativas, los criterios, los expertos y el conjunto de términos lingüísticos que utilizarán para emitir sus valoraciones, las cuales se recopilan en la segunda fase. En la última fase se pueden emplear los operadores de agregación para 2-tuplas, media aritmética, media aritmética ponderada, media geométrica, media geométrica ponderada y media ponderada ordenada. Además se presentan un grupo de funcionalidades que brindan valor agregado al sistema tales como un módulo de registro de usuarios para garantizar la seguridad, un sistema de notificaciones por correo, un campo de descripción del problema y la posibilidad de exportar el problema antes y después de tener su solución. La arquitectura y tecnologías empleadas facilitan la incorporación de nuevas funcionalidades, con lo que se obtiene una herramienta flexible, escalable y adaptable para todos los entornos.

Palabras clave: 2-tupla lingüística, computación con palabras, modelo computacional, problema Toma de Decisiones, Sistema de Apoyo a la Toma de Decisiones.

Abstract

The 2-tuple linguistic model is widely used in solving linguistic decision-making problems (LDM) because it offers close to natural human language results and provides a high level of interpretability while allowing the development of processes Computing with Words without loss of information. With the aim of supporting the linguistic decision-making problem based on the 2-tuple linguistic model, the XABAL 2 TUPLE SOLVE web system was developed. The system solves LDM problems through a resolution scheme based on decision analysis, which includes three main phases: Definition of the Problem Framework, Data Collection and Aggregation of Information. In the first phase the elements of the problem are defined: the alternatives, criteria, experts and the set of linguistic terms used to issue their valuations, which are collected in the second phase. In the last phase can be used aggregation operators for 2-tuples, arithmetic mean, weighted arithmetic mean, geometric mean, weighted mean and geometric weighted average orderly. Additionally, a group of features that provide added value to the system such as a user registration module to ensure security, a notifications system by e-mail, a description field of the problem and the possibility of exporting the problem before and after getting a solution. Technologies and architecture used, facilitate the implementation of new functionalities so a flexible, scalable, customizable tool for all environments is obtained.

Key words: 2-tuple linguistic, computing with words, computational model, decision-making problem, decision support system.

Índice general

Introducción	1
1. Problemas de Toma de Decisión Lingüística y soluciones basadas en el Modelo Computacional 2-tupla Lingüística	6
1.1. Problemas de Toma de Decisiones	6
1.2. Enfoque Lingüístico Difuso	9
1.3. Computación con Palabras	12
1.4. Modelo Computacional 2-Tupla Lingüística	13
1.5. Esquema de resolución de un Problema de Toma de Decisiones Lingüístico basado en el Modelo Computacional 2-Tupla Lingüística	18
1.6. Sistema de Apoyo a la Toma de Decisiones	20
1.7. Análisis de Sistemas de Apoyo a la Toma de Decisiones que utilizan el Modelo 2-Tupla Lingüística para el tratamiento de información lingüística	21
1.8. Conclusiones parciales	22
2. Descripción del sistema XABAL 2 TUPLE SOLVE	24
2.1. Herramientas y Tecnologías	24
2.1.1. Metodología de desarrollo de software	24
2.1.2. Lenguaje de modelado	26
2.1.3. Herramienta de modelado	26
2.1.4. Lenguaje de programación	27
2.1.5. Entorno de desarrollo integrado	27
2.1.6. Marco de trabajo	27
2.1.7. Sistema gestor de bases de datos	28
2.1.8. Intercambio con el servidor	29
2.1.9. Servidor Web	29
2.1.10. Bibliotecas	29
2.2. Modelo de dominio	32
2.2.1. Diagrama de clases del dominio	32
2.2.2. Descripción del modelo de dominio	32
2.2.3. Definición de las clases del modelo de dominio	32
2.3. Descripción de la propuesta de solución	33

2.4.	Requisitos del sistema	34
2.4.1.	Requisitos funcionales	34
2.4.2.	Requisitos no funcionales	35
2.5.	Modelo de Casos de Uso del Sistema (CUS)	37
2.5.1.	Descripción de los actores que interactúan con el sistema.	37
2.5.2.	Identificación de los CUS	37
2.5.3.	Diagrama de Casos de Uso del Sistema	38
2.5.4.	Descripción extendida de los CUS	38
2.6.	Conclusiones parciales	40
3.	Implementación y pruebas del sistema XABAL 2 TUPLE SOLVE	41
3.1.	Arquitectura de Software	41
3.1.1.	Patrones arquitectónicos	41
3.2.	Modelo de diseño	42
3.2.1.	Diagrama de paquetes	42
3.2.2.	Diagrama de Clases del Diseño	43
3.2.3.	Patrones de Diseño	44
3.3.	Modelo de datos	46
3.4.	Modelo de implementación	47
3.5.	Modelo de despliegue	47
3.6.	Algoritmos implementados	48
3.7.	Verificación y validación	49
3.8.	Pruebas de Software	50
3.8.1.	Pruebas de caja negra	50
3.9.	Pruebas de rendimiento	53
3.9.1.	Pruebas de carga	53
3.10.	Caso de estudio	54
3.11.	Comparación FLINTSTONES – XABAL 2 TUPLE SOLVE	57
3.12.	Conclusiones parciales	59
	Conclusiones	61
	Recomendaciones	62
	Bibliografía	66

Índice de figuras

1.1. Clasificación de los Problemas de Toma de Decisiones.	7
1.2. Esquema de resolución general de un Problema de Toma de Decisión	8
1.3. Proceso de Toma de Decisiones.	9
1.4. Esquema lingüístico para manejar la incertidumbre.	13
1.5. Traslación simbólica y representación en 2-tupla lingüística.	15
1.6. Proceso de resolución de un problema de Toma de Decisiones Lingüístico.	18
1.7. Esquema de resolución de un problema de Toma de Decisiones Lingüístico basado en el modelo computacional 2-tupla lingüística	18
2.1. Diagrama de clases del dominio	32
2.2. Diagrama de Casos de Uso del Sistema	38
3.1. Diagrama de paquetes	42
3.2. Diagrama de Clases del Diseño: Aplicar Operador	44
3.3. Diagrama Entidad-Relación	46
3.4. Diagrama de Implementación: Aplicar Operador	47
3.5. Diagrama de despliegue	48
3.6. Código del algoritmo media aritmética.	48
3.7. Código del algoritmo media geométrica.	49
3.8. Código del algoritmo media aritmética ponderada.	49
3.9. Resultados de las pruebas de Caja Negra	53
3.10. Prueba de carga de XABAL 2 TUPLE SOLVE	54
3.11. Representación gráfica del conjunto de términos lingüísticos de siete términos	55
3.12. Gestión de información	55
3.13. Valoración de los expertos	56
3.14. Salida del sistema	57
3.15. Resultado del caso de estudio mediante el uso de FLINTSTONES	57

Índice de tablas

2.1. Descripción de los actores que interactúan con el sistema	37
2.2. Casos de uso del sistema	37
2.3. Descripción del CUS Aplicar Operador	39
3.1. Casos de prueba del CUS Aplicar Operador	52
3.2. Variables del CUS Aplicar Operador	52
3.3. Evaluaciones proporcionadas por los expertos.	56
3.4. Comparación FLINTSTONES – XABAL 2 TUPLE SOLVE	59

Introducción

La Toma de Decisiones es un proceso habitual en muchas actividades del mundo real como la ingeniería, la economía, la psicología, entre otros. La Toma de Decisiones se define como el “proceso de definición de problemas, recopilación de datos, generación de alternativas y selección de un curso de acción” (Hellriegel et al., 2002). Los Problemas de Toma de Decisiones por tanto, están definidos en contextos diversos, por ejemplo, pueden describir situaciones bajo incertidumbre, donde los humanos toman sus decisiones con conocimiento incierto y/o basado en sus propias percepciones (Rodríguez et al.).

En ocasiones la modelación de los Problemas de Toma de Decisiones se torna muy compleja debido a que estos pueden contener información con un alto grado de incertidumbre, además de intentar representar aspectos cualitativos asociados a algún tipo de valoración. Por otra parte, usualmente se trabaja en entornos cuantitativos donde la información se expresa por medio de valores numéricos. Sin embargo, muchos aspectos del mundo real no pueden ser valorados de esta forma, aunque sí mediante percepciones o con conocimiento impreciso. En estos casos el Enfoque Lingüístico Difuso (Zadeh, 1975) modela dicho conocimiento mediante información lingüística por lo que se denomina Toma de Decisión Lingüística (TDL). Los resultados satisfactorios del empleo del Enfoque Lingüístico Difuso para la resolución de problemas de Toma de Decisión Lingüística, se debe, a que los conjuntos difusos son adecuados para tratar la incertidumbre por la cercanía de la información lingüística al modelo cognitivo humano.

El uso de información lingüística implica la necesidad de realizar procesos de Computación con Palabras (Bonissone and Decker, Zadeh, 1996a), estrategia que incluye técnicas computacionales que definen diversas operaciones tales como: agregación, negación y comparación, sobre información lingüística.

Se han desarrollado diversos modelos para realizar procesos de Computación con Palabras, necesarios en los modelos de resolución de problemas de TDL. Los más importantes son el “Modelo basado en el principio de extensión” que opera sobre los números difusos que definen la semántica de los términos lingüísticos (Degani and Bortolan, 1988, Tong and Bonissone, 1980); el “Modelo simbólico” que considera una escala ordinal de etiquetas, para realizar cálculos sobre los índices de los términos lingüísticos (Buckley, 1984, Yager, 1981) y el “Modelo de representación lingüística 2-tupla ” que propone el concepto de traslación simbólica y opera en un dominio de expresión lingüístico pero tratándolo como un universo continuo, para así aumentar la precisión en los resultados. Para algunos autores el modelo 2-tupla lingüística es considerado un modelo simbólico (Herrera and Martínez, 2001, Wei, 2010). Junto a este modelo de representación de información lingüística, Herrera y Martínez definieron un modelo computacional lingüístico basado en dos funciones de transformación que permiten convertir valores numéricos a 2-tupla y viceversa, sin pérdida de información, por lo que numerosos operadores de agregación numéricos tradicionales y otros, han sido extendidos a 2-tuplas. Las múltiples ventajas y amplio desarrollo teórico del modelo, no se corresponden con el desarrollo práctico pues hasta donde se conoce

solamente una herramienta soporta la solución de problemas con este enfoque e incluye un número limitado de operadores de agregación.

Un Sistema de Apoyo a la Toma de Decisiones es un “sistema de información basado en un computador interactivo, flexible y adaptable, especialmente desarrollado para apoyar la solución de un problema de gestión no estructurado para mejorar la Toma de Decisiones. Utiliza datos, proporciona una interfaz amigable y permite la Toma de Decisiones en el propio análisis de la situación” (Turban et al., 2005). El objetivo general debe centrarse en darle al usuario las alternativas más factibles basándose en la información proporcionada por este y realizar las operaciones pertinentes de una forma sencilla para eliminar la necesidad de tener amplios conocimientos sobre el tema.

La actual evolución y complejidad de los problemas de Toma de Decisiones, donde predominan las operaciones con conjuntos en lenguaje natural, han puesto de manifiesto la necesidad de un complemento que apoye la resolución de los mismos. Los estudios reflejan entre algunos aspectos: problemas de centralización de la información, porque los expertos de un tema pueden tener diferentes ubicaciones geográficas, falta de comunicación, comprensión, conocimiento y más de una estrategia de solución que demuestre, que es acertada la propuesta que se obtuvo. Esto indicaría la utilización de otros operadores que cumplan el mismo propósito, tarea que implicaría un alto costo temporal no sólo en la búsqueda de la primera solución sino también de las restantes. Estas razones demuestran el beneficio que traería consigo la creación de una herramienta capaz de hacer más ágil y transparente este proceso para la sociedad, debido a las aplicaciones de todos estos conceptos en cada una de las empresas existentes a niveles internacionales.

En este punto, se ha identificado una contradicción: por sus resultados satisfactorios, el modelo lingüístico 2-tupla es ampliamente utilizado en la resolución de problemas de TDL en tanto ofrece resultados cercanos al lenguaje humano natural y provee un alto nivel de interpretabilidad a la vez que permite implementar procesos de Computación con Palabras de forma precisa al mejorar el proceso de retransformación. Sin embargo, a pesar del desarrollo teórico del modelo, de los avances en el campo de los Sistemas de Apoyo a la Toma de Decisiones y las ventajas que su aplicación presupone en la solución de problemas de TDL con múltiples expertos y múltiples criterios, no hay evidencias de grandes avances en el uso de este tipo de soluciones para este tipo de problemas.

Las ideas anteriores permiten identificar el siguiente **problema de investigación**: ¿Cómo contribuir a la resolución de problemas de Toma de Decisión Lingüística basada en el modelo 2-tupla lingüística?

La existencia del problema antes mencionado determina como **objeto de estudio** la Toma de Decisión Lingüística y delimita el **campo de acción** hacia los Sistemas de Apoyo a la Toma de Decisiones con información representada en 2-tupla lingüística. El **objetivo general** es desarrollar un Sistema de Apoyo a la Toma

de Decisiones que utilice el modelo 2-tupla lingüística para dar solución a los problemas de Toma de Decisión Lingüística. Este propósito general implica el cumplimiento de los siguientes **objetivos específicos**:

- Realizar un estudio del estado del arte sobre los procesos y Problemas de Toma de Decisiones con información representada en lenguaje natural.

Para ello se analizan en profundidad los elementos del proceso de Toma de Decisiones. Se estudian los Problemas de Toma de Decisiones con información lingüística y los métodos más representativos para darle solución. Esto permite delimitar las dificultades existentes, de manera que se puedan identificar las limitaciones actuales.

- Analizar las herramientas y modelos computacionales que den solución a Problemas de Toma de Decisiones con información lingüística mediante el modelo computacional 2-tupla lingüística.

Se estudian las herramientas que implementan el modelo computacional 2-tupla lingüística para resolver Problemas de Toma de Decisiones. Razón que permite reconocer las ventajas que presentan los sistemas que existen y realizar una estrategia que cubra las necesidades que estos no solucionan. Se estudian además otros modelos que tengan un funcionamiento similar al modelo computacional 2-tupla lingüística.

- Desarrollar un sistema que permita la solución de Problemas de Toma de Decisiones con información lingüística basado en el uso del modelo computacional 2-tupla lingüística y sus operadores de agregación.

Para dirigir correctamente el proceso investigativo se definen las siguientes **preguntas de investigación**:

1. ¿Cómo se define el proceso de Toma de Decisiones?
2. ¿Qué elementos componen un problema de Toma de Decisiones?
3. ¿Qué es un Sistema de Apoyo a la Toma de Decisiones?
4. ¿Por qué es necesaria la representación de información lingüística?
5. ¿Qué permite la computación con palabras?
6. ¿Cuáles son los modelos computacionales existentes?
7. ¿Cómo funciona el modelo computacional 2-tupla lingüística?
8. ¿Qué es un operador de agregación?
9. ¿Qué operadores de agregación contiene el modelo computacional 2-tupla lingüística?
10. ¿Cómo resolver un Problema de Toma de Decisiones con información lingüística mediante el modelo computacional 2-tupla lingüística y sus operadores?

11. ¿Cómo diseñar y modelar un sistema interactivo, flexible y adaptable que utilice el modelo computacional 2-tuplas para la resolución de Problemas de Toma de Decisiones?
12. ¿Cómo implementar un Sistema de Apoyo a la Toma de Decisiones que garantice la mejor respuesta a la hora de su utilización?

Para dar cumplimiento al objetivo general se proponen las siguientes **tareas**:

1. Análisis de la resolución de Problemas de Toma de Decisiones para conocer su estructura y funcionamiento.
2. Comprobación de los Sistemas de Apoyo a la Toma de Decisiones existentes para comprender sus estrategias de trabajo.
3. Comprensión de la representación de información lingüística para explotar sus ventajas.
4. Análisis de la Computación con Palabras como concepto para su posterior aplicación.
5. Verificación de los modelos computacionales actuales para conocer las posibles tendencias.
6. Comprobación de ventajas del modelo computacional 2-tupla lingüística para fundamentar su superioridad en relación a otros modelos.
7. Análisis y comparación de los operadores de agregación para seleccionar los más utilizados por la comunidad científica.
8. Fundamentación de la metodología de software, las herramientas y tecnologías a utilizar en el proceso de desarrollo.
9. Modelado de los requisitos funcionales y no funcionales de la propuesta de solución para definir sus características.
10. Diseño de la propuesta de solución para guiar el proceso de implementación.
11. Implementación de la propuesta de solución para dar cumplimiento al objetivo planteado.
12. Validación de la propuesta de solución a través de pruebas para demostrar su correcto funcionamiento.

Para darle cumplimiento a las tareas anteriores se utilizarán los siguientes **métodos de la investigación científica**:

Métodos teóricos:

Se basan en la utilización del pensamiento en sus funciones de deducción, análisis y síntesis (Barchini, 2005). En la presente investigación los métodos teóricos a utilizar son:

- **Método analítico - sintético:** Es utilizado para consultar la bibliografía especializada referente al tema e identificar elementos claves que contribuyan a la solución del problema científico planteado, permite sintetizar conceptos que ayudarán a comprender la solución del problema.

- **Método sistémico:** Utilizado en el estudio de la herramienta FLINTSTONES para comprender su estructura, forma de trabajo y complejidad, con el objetivo de definir la forma de implementación que se debe llevar a cabo en el desarrollo del sistema propuesto.

Métodos empíricos:

Describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Aunque existen diversas opiniones la mayoría de los autores concuerdan que los métodos empíricos generales son: la observación, la medición y la experimentación (León and González, 2002).

- **Método de observación selectiva:** Utilizado en el estudio de la herramienta FLINTSTONES para conocer cómo trabaja en el tratamiento de información lingüística mediante el uso del modelo computacional 2-tupla lingüística.

Estructura del documento de tesis:

El documento de tesis está estructurado en tres capítulos donde se presenta la información necesaria para poder comprender la investigación en su totalidad.

El capítulo 1, titulado “Fundamentos teóricos sobre los Sistemas de Apoyo a la Toma de Decisiones”, presenta los conceptos asociados al tema para crear la base teórica de la investigación. Incluye el estado actual de desarrollo de los Sistemas de Apoyo a la Toma de Decisiones al hacer un estudio de los ya existentes.

En el capítulo 2 se realiza un estudio que permite seleccionar las herramientas necesarias para desarrollar un sistema que cubra las deficiencias que se muestran en la problemática descrita. Además se presenta una modelación del dominio del problema, se plantean los requisitos funcionales y no funcionales, así como los artefactos generados en el modelo del sistema, estos elementos se enmarcan bajo el título “Descripción del sistema XABAL 2 TUPLE SOLVE”.

Las conclusiones de la sección anterior permiten tener los conocimientos necesarios para abordar el capítulo 3 que se titula “Implementación y prueba del sistema XABAL 2 TUPLE SOLVE” y está conformado por la definición de las clases de diseño, sus relaciones y la estructuración de los diagramas de clases del diseño. Se presenta el modelo de datos, diagrama de despliegue y de implementación y concluye con la validación del sistema a partir de las pruebas necesarias para demostrar que la solución cumple con los objetivos trazados en la investigación.

Capítulo 1

Problemas de Toma de Decisión Lingüística y soluciones basadas en el Modelo Computacional 2-tupla Lingüística

En este capítulo se revisan conceptos asociados al objeto y el campo de estudio de la investigación, que constituyen el marco conceptual de los resultados de este trabajo. Primero se realiza un acercamiento a los problemas de Toma de Decisiones con especial interés en la Toma de Decisiones Lingüística. Luego se analizan el Enfoque Lingüístico Difuso y la Computación con Palabras debido a su necesidad para resolver este tipo de problemas. Se estudia del Modelo Computacional 2-Tupla Lingüística, como elemento principal de la investigación y la forma de resolver problemas de Toma de Decisiones Lingüística mediante el uso de este. Se analiza el concepto y herramientas de Apoyo a la Toma de Decisiones que utilizan el modelo.

1.1. Problemas de Toma de Decisiones

Para lograr una mejor comprensión de un tema, siempre será necesario definirlo según los criterios generales que aportan aquellas investigaciones referentes al mismo. Por esta razón se hace a continuación un análisis de algunos conceptos relacionados con los Problemas de Toma de Decisiones.

Un Problema de Toma de Decisiones es el “proceso de definición de problemas, recopilación de datos, generación de alternativas y selección de un curso de acción” (Hellriegel et al., 2002).

Para (Quesada, 2015) la Toma de Decisión es una tarea habitual de la vida cotidiana. Constantemente se enfrentan situaciones en las que existen varias alternativas y, al menos en algunas ocasiones, se necesita decidir cuál es mejor o cuál llevar a cabo.

Los autores de esta investigación asumen el primer concepto mencionado que engloba todo el proceso de Toma de Decisión y define en etapas claras las tareas que se deben realizar.

Los Problemas de Toma de Decisiones se pueden clasificar según el ambiente de decisión, número de criterios, número de expertos, consideración de los cambios en el tiempo y dominios de expresión. En esta incorporación se profundizará en número de criterios y número de expertos, específicamente en los problemas multicriterio y multiexperto. En la Figura 1.1 se muestran las clasificaciones mencionadas y se encierra entre líneas discontinuas las utilizadas, junto a una síntesis de su funcionamiento.

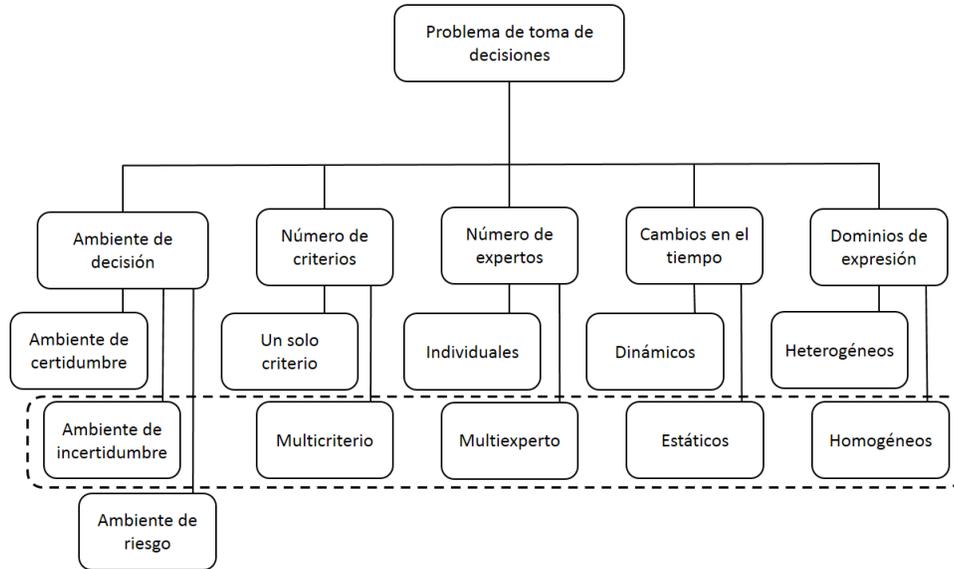


Figura 1.1: Clasificación de los Problemas de Toma de Decisiones.

Ambiente de Decisión:

- Ambiente de Certidumbre: Un problema de decisión está definido en un ambiente de certidumbre si se conocen con exactitud todos los elementos o factores que intervienen en el problema.
- Ambiente de Riesgo: Un problema de decisión está definido en un ambiente de riesgo si alguno de los elementos o factores que intervienen están sujetos a las leyes del azar.
- Ambiente de Incertidumbre: Un problema de decisión está definido en un ambiente de incertidumbre si la información disponible sobre las distintas alternativas puede ser incompleta, vaga o imprecisa, lo que implica que la utilidad asignada a cada alternativa tenga que ser valorada de forma cualitativa.

Número de Criterios:

- Problemas con un sólo criterio o atributo: Problemas de decisión en los que, para evaluar las alternativas, se tiene en cuenta un único criterio o atributo que representa la valoración dada a esa alternativa.
- Problemas multicriterio o multiatributo: Problemas de decisión en los que, para evaluar las alternativas, se tienen en cuenta dos o más criterios o atributos que definen cada alternativa.

Número de Expertos:

- Unipersonales o Individuales: Las decisiones son tomadas por un único experto. En los problemas de decisión unipersonal o individual, cada alternativa es valorada por un único experto.
- En Grupo o Multiexperto: Las decisiones son tomadas en conjunto por un grupo de expertos que intentan alcanzar una solución, en común, al problema.

Consideración de los Cambios en el Tiempo:

- Estáticos: El punto de vista tradicional o clásico de la toma de decisión. Se tienen en cuenta las preferencias emitidas en un único momento de decisión sobre un único conjunto de alternativas y de acuerdo a un único conjunto de criterios de importancia fija.
- Dinámicos: Se tienen en cuenta múltiples momentos de decisión, ya sea para tomar una única decisión al final del proceso o para tomar decisiones en cada uno de los múltiples momentos. Las alternativas y criterios valorados no son necesariamente fijos. El comportamiento temporal de las alternativas influirá en las decisiones.

Dominios de Expresión:

- Homogéneos: Todas las preferencias son emitidas con el mismo dominio de información.
- Heterogéneos o no-homogéneos: Las preferencias son emitidas con más de un dominio de información.

Independiente al tipo de problema de decisión que se intente resolver, el esquema básico de resolución de estos siempre tendrá presentes las dos fases que se muestran en la siguiente Figura 1.2.



Figura 1.2: Esquema de resolución general de un Problema de Toma de Decisión

- **Agregación:** Obtener un valor colectivo de preferencias para cada alternativa y/o criterio de representación, a partir de los valores individuales de preferencias proporcionados por los expertos, se utiliza el operador de agregación adecuado a las necesidades del problema.
- **Explotación:** Seleccionar la/s mejor/es alternativa/s a partir de los valores colectivos y la definición de un criterio de selección que permita establecer un orden de mérito entre el conjunto de alternativas al problema.

En condiciones reales, existen situaciones complejas que dificultan que los decisores lleguen a las fases de agregación y explotación de forma directa. En estos casos, se suele recurrir a un proceso general de Toma de Decisiones que puede llegar a tener ocho fases, de las cuales las siete primeras pertenecen al análisis de decisión y la última la toma de la decisión. Según las particularidades de las problemáticas y los objetivos que intenten cumplir, se pueden redefinir estas fases o analizar parte de ellas. Para esta contribución se analizaron solo tres fases del proceso general, las cuales se mencionan y explican a continuación de la Figura 1.3:

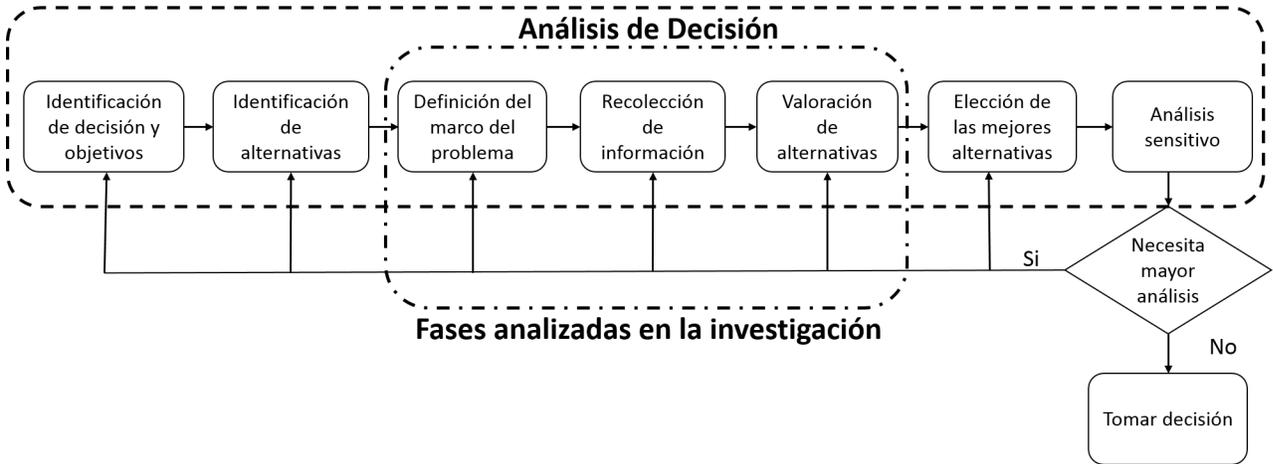


Figura 1.3: Proceso de Toma de Decisiones.

- Definición del marco del problema: Se plantean todos los elementos que componen el problema: criterios, expertos, variables y conjunto de términos lingüísticos.
- Recopilación de información: Se investiga el tema en cuestión mediante el uso de los criterios existentes sin perder de vista las metas deseadas.
- Valoración: Se presentan el conjunto de alternativas o variantes por cada experto basadas en el uso de los criterios especificados.

Al modelar un Problema de Toma de Decisiones es esencial el modelado de las preferencias, es decir, la definición de una representación concreta con la cual los expertos expresan sus valoraciones. Para ello se tiene en cuenta el conocimiento de los expertos acerca del problema, sus opiniones personales y experiencias, además de la naturaleza de las alternativas o los criterios que las definen. Este aspecto, como se explicó, determina si un problema es heterogéneo u homogéneo. El modelado lingüístico es adecuado si la información disponible es demasiado imprecisa para justificar el uso de valores numéricos o si la imprecisión de la información puede ser aprovechada para alcanzar robustez, solución a bajo coste y una buena interpretación de la realidad. Los expertos que resuelven Problemas de Toma de Decisiones en ambientes de incertidumbre, usualmente se sienten más cómodos al realizar valoraciones mediante términos lingüísticos debido a que es el modo en que se comunican, dando lugar a la TDL. A continuación se analizan el Enfoque Lingüístico Difuso (Zadeh, 1975) y la Computación con Palabras por su aplicabilidad en problemas de TDL (Bonissone, 1980, Degani and Bortolan, 1988, Herrera and Martínez, 2000, Torra, 2001, Xu, 2004).

1.2. Enfoque Lingüístico Difuso

Enfoque Lingüístico Difuso (Zadeh, 1975) proporciona una representación directa de los aspectos de naturaleza cualitativa mediante variables lingüísticas (Herrera et al., 2000).

El Enfoque Lingüístico Difuso (Zadeh, 1975) modela el conocimiento mediante información lingüística teniendo como soporte los conjuntos difusos. La Lógica Difusa es una de las propuestas surgidas para la formalización del razonamiento aproximado, que intenta manejar el conocimiento propio del sentido común. Se trata de una generalización de la lógica booleana clásica, propuesta por Zadeh en (Zadeh, 1965) como una extensión de esta, con el objetivo de permitir manejar el concepto de verdades parciales situadas entre el “completamente verdadero” y el “completamente falso”, pues la percepción del mundo real para el ser humano está invadida por conceptos que no tienen fronteras nítidamente definidas (Zadeh, 1996b).

Una variable lingüística se caracteriza por un valor sintáctico o etiqueta y por un valor semántico o significado. La etiqueta es una palabra que pertenece a un Conjunto de Términos Lingüísticos (CTL) y su significado viene dado por un subconjunto difuso en un universo del discurso. Dado que las palabras son menos precisas que los números, el concepto de variable lingüística parece ser adecuado para caracterizar objetos que son demasiado complejos o no están completamente definidos para poder ser evaluados mediante valores numéricos precisos. Una variable lingüística en el Enfoque Lingüístico Difuso es definida como sigue:

Definición según (Herrera et al., 2000): Una variable lingüística se caracteriza por una quintupla $(H, T(H), U, G, M)$, donde:

- H es el nombre de la variable;
- $T(H)$ (o sólo T) simboliza el conjunto de términos de H , es decir, el conjunto de nombres de valores lingüísticos de H ;
- U es el universo del discurso;
- G es una regla sintáctica (que normalmente toma forma de gramática) para generar los nombres de los valores de H ; y
- M es una regla semántica para asociar significado $M(X)$, a cada elemento de H , el cual es un conjunto difuso de U .

La resolución de un problema bajo la perspectiva del Enfoque Lingüístico Difuso, cuenta con dos operaciones básicas y fundamentales:

- Elección de un adecuado CTL, $T(H)$.
- Definición de la semántica asociada a cada término lingüístico.

El objetivo de establecer los términos lingüísticos de una variable lingüística es proporcionar a una fuente de información un número adecuado de términos con los cuales se pueda expresar con facilidad una opinión o conocimiento. Para alcanzar este objetivo se analiza la granularidad de la incertidumbre, que es la cardinalidad

del CTL definido. Un CTL tiene una granularidad baja o un tamaño de grano grueso cuando su cardinalidad es pequeña. Esto significa que el dominio está poco particionado y que existen pocos niveles de distinción de la incertidumbre. Cuando la cardinalidad del CTL es alta, entonces tiene una granularidad alta o un tamaño de grano fino. Esta granularidad es adecuada cuando el conocimiento de los expertos sobre la variable a valorar es alto.

Existen diferentes formas de seleccionar los términos lingüísticos apropiados del conjunto de términos y su semántica. Los principales enfoques para seleccionar los términos lingüísticos son:

1. Enfoque de Estructura Ordenada: Se define el término lingüístico establecido por medio de una estructura ordenada que proporciona el conjunto de términos, S , distribuido en una escala en la que se define un orden total (Yager, 1995). Por ejemplo, un conjunto de siete términos S , se podría dar de la siguiente manera gráficamente, ver Figura 1.5.

$$S = \{s_0 : nada(n), s_1 : muy\ bajo(mb), s_2 : bajo(b), s_3 : medio(m), s_4 : alto(h), s_5 : muy\ alto(ma), s_6 : perfecto(p)\}$$

Por lo general, en estos casos, es necesario el uso de los siguientes operadores:

- Un operador de negación
- Un operador de maximización
- Un operador de minimización

2. Enfoque de Gramática Libre de Contexto: Se define el término lingüístico establecido por medio de una gramática libre de contexto, G , de tal manera que los términos lingüísticos son frases generadas por G (Bonissone, 1980, Rodríguez et al., 2012, Zadeh, 1975).

A continuación se explican tres variantes que se pueden utilizar para la definición de la semántica de un CTL:

1. Semántica basada en funciones de pertenencia y una regla semántica: Este enfoque supone que el significado de cada término lingüístico está dado por medio de un subconjunto difuso definido en el intervalo de $[0,1]$, que se describe por funciones de pertenencia (Bordogna and Pasi, 1993). Se utiliza este enfoque cuando la semántica de los términos lingüísticos se crea por medio de una gramática generadora. Así, se establece por medio de dos elementos (Bonissone, 1980, Zadeh, 1975):

- a) Los conjuntos difusos primarios asociados con los términos lingüísticos primarios.
- b) Una regla semántica M para la generación de los conjuntos difusos, a partir de los conjuntos difusos no primarios.

2. Semántica basada en una estructura ordenada del conjunto de términos lingüísticos: Esta alternativa presenta la semántica de la estructura definida sobre el conjunto de términos lingüísticos. Esto sucede cuando los usuarios proporcionan sus evaluaciones mediante el uso de un conjunto ordenado de términos lingüísticos. Bajo esta semántica acercarse a la distribución de los términos lingüísticos en escala de $[0,1]$ puede ser igualmente informativo (simétrico) (Yager, 1995) o no (no simétrico) (Herrera et al., 2008, Torra et al., 1996).
3. Semántica mixta: Todos los términos lingüísticos se consideran términos primarios. Asume elementos de los enfoques anteriores, es decir, una estructura ordenada de los términos lingüísticos primarios y los conjuntos difusos de semántica de los términos lingüísticos. Al igual que en la semántica basada en una estructura ordenada, se asume que los conjuntos ordenados de términos lingüísticos se distribuyen en una escala suponiendo que cada término lingüístico es igualmente informativo. Por otra parte, como en la semántica basada en funciones de pertenencia y una regla semántica, se define la semántica de los términos lingüísticos primarios por medio de los conjuntos difusos (Delgado et al., 1998).

1.3. Computación con Palabras

El uso de variables lingüísticas implica procesos de Computación con Palabras o Computing with Words, tales como agregación y comparación. La Computación con Palabras es una metodología que permite el razonamiento o Toma de Decisiones con información en lenguaje natural. Zadeh en (Mendel et al., 2010) plantea que esta razón le brinda mayor importancia porque así se representa gran parte del conocimiento humano, además de que este permite un nivel de imprecisión sin incluir errores, el resultado siempre corresponde a una etiqueta según su grado de pertenencia. Las características de esta metodología le permiten que sea utilizada en diversas áreas de investigación, con el fin de minimizar la pérdida de información al realizar los procesos de razonamiento y cálculo (Herrera et al., 2009).

Para Zadeh la computación con palabras es una metodología de razonamiento, computación y toma de decisiones en la que se utilizan palabras procedentes del lenguaje natural. Ha sido aplicada como base computacional en toma de decisiones lingüística (Herrera et al., 2009, Martínez et al., 2007), ya que proporciona herramientas cercanas a los procesos de razonamiento de los seres humanos relacionados con la toma de decisiones.

Esta estrategia persigue tres objetivos fundamentales. En primer lugar, ofrecer una metodología para calcular y razonar en caso de que la información disponible no sea lo suficientemente precisa como para justificar el empleo de números. En segundo lugar, aprovechar la tolerancia de la imprecisión para alcanzar manejabilidad, robustez, bajo coste y mejor relación con la realidad. Y, en tercer lugar, proporcionar bases para el desarrollo de lenguajes de programación que pudieran aproximarse a los lenguajes naturales en apariencia

y en capacidad de expresión (Zadeh, 1996b).

Yager define que el objetivo de la computación con palabras es incluir la fuente de información más común para los seres humanos en una computadora, formalizándola mediante modelos basados en la Toma de Decisiones. Este proceso tendría tres fases: transformación, manipulación y retransformación que permitirían aumentar la comprensión de los resultados al mostrar una solución más evidente (Mendel et al., 2010).

Entre las incorporaciones de Yager se encuentran, propuestas de esquemas lingüísticos para manejar la incertidumbre como el que se muestra en la Figura 1.4. (Yager, 2004, Yager and Kreinovich, 1999). En estos, Yager señala la importancia de las fases de transformación y retransformación en los procesos de computación con palabras. La primera implica transformar la información lingüística en un formato manipulable por las máquinas basado en herramientas difusas, y la segunda, consiste en convertir los resultados computacionales en información lingüística para facilitar su comprensión.

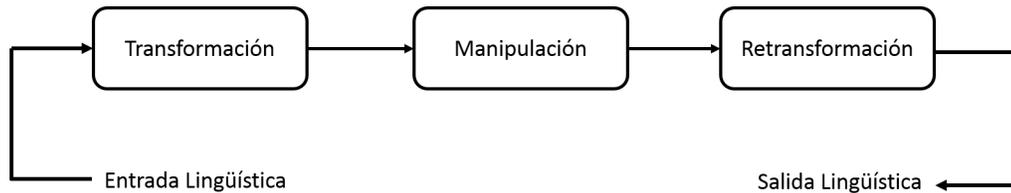


Figura 1.4: Esquema lingüístico para manejar la incertidumbre.

Los conceptos analizados presentan muchos puntos comunes debido a que independientemente al autor, para realizar los procesos de computación con palabras se debe cumplir sus tres componentes esenciales.

- Una palabra debe conducir a una función de pertenencia en lugar de una función de pertenencia conduzca a una palabra.
- Los números por sí solos no pueden activar el motor de computación con palabras.
- La salida de la computación con palabras debe ser por lo menos una palabra y no sólo un número.

Se decide utilizar el concepto propuesto por Zadeh, pues este define el contenido de una forma más precisa. Además las ideas de Yager prácticamente se alimentan de las bases que el otro autor brinda. No se dejarán de analizar apreciaciones puntuales del segundo autor como es el caso de sus propuestas de diagramas.

1.4. Modelo Computacional 2-Tupla Lingüística

El modelo computacional 2-tupla lingüística sigue el paradigma de computación con palabras. Este modelo proporciona la precisión, simplicidad y la interpretación de los cálculos en un conjunto de términos lingüísticos con un valor impar de granularidad. Dichas funciones se expresan de forma triangular, simétrica y uniforme-

mente distribuida en el intervalo. El modelo 2-tupla lingüística se crea con el objetivo de disminuir la pérdida de información de mínima a nula a la hora de realizar una operación. Estas razones destacan su selección por ser la más idónea entre las alternativas de modelos computacionales simbólicos.

El modelo simbólico 2-tupla fue introducido por Herrera y Martínez en (Herrera and Martínez, 2000) con el objetivo de mejorar la precisión de los resultados y facilitar los procesos de Computación con Palabras, trata el dominio lingüístico como un dominio continuo pero mantiene la base lingüística (sintaxis y semántica). Para ello, este modelo extiende la representación lingüística difusa que añade un nuevo parámetro.

El modelado de la información lingüística está basado en el concepto de traslación simbólica y lo utiliza para representar la información lingüística mediante un par de valores, llamados 2-tupla lingüística (s_i, α) , donde $s_i \in S = \{s_0, \dots, s_g\}$ es un término lingüístico y α es un valor numérico representando la traslación simbólica.

1. Sea $s_i \in S$ un término lingüístico similar al Enfoque Lingüístico Difuso cuya semántica se define por una función de pertenencia difusa y la sintaxis elegida en función de las opciones que ofrece el Enfoque Lingüístico Difuso,
2. α es un valor numérico, traslación simbólica, que indica la traslación de la función de pertenencia difusa que representa el término más cercano $s_i \in S$ si s_i no coincide con la información lingüística computarizada. El valor de α se define entonces como (Martínez et al., 2015):

$$\alpha = \begin{cases} [-0.5, 0.5) & \text{si } s_i \in \{s_1, \dots, s_{g-1}\} \\ [0, 0.5) & \text{si } s_i = s_0 \\ [-0.5, 0) & \text{si } s_i = s_g \end{cases}$$

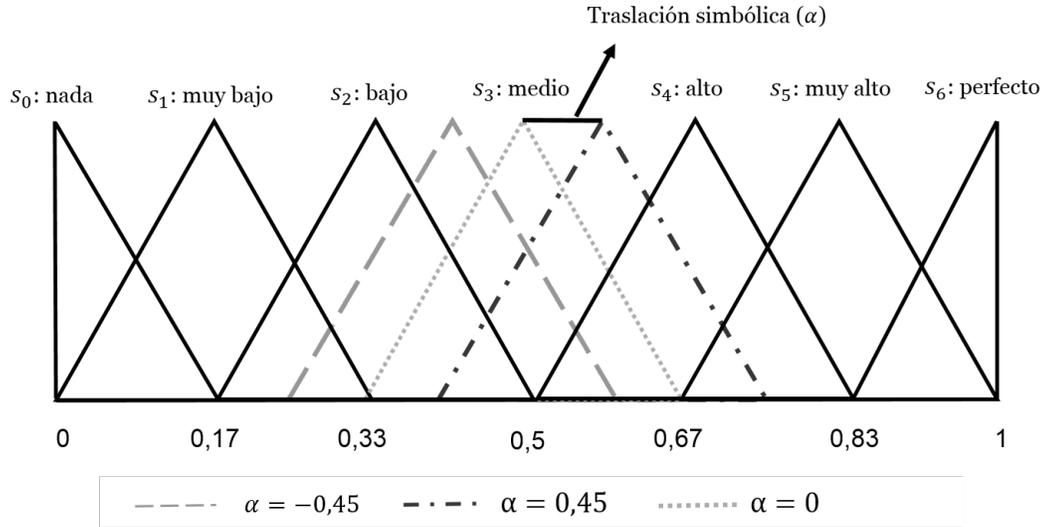


Figura 1.5: Traslación simbólica y representación en 2-tupla lingüística.

La información lingüística expresada por este par de elementos se observa como (s_i, α) y el valor lingüístico 2-tupla. Para una mejor comprensión de esta representación se debe tener en cuenta que un cálculo simbólico en términos lingüísticos en S se obtiene un valor $\beta \in [0, g] \in \mathfrak{R}$ que se transformará en un valor equivalente en 2-tupla lingüística, (s_i, α) por medio de la función Δ_s que se define a continuación.

La 2-tupla asociada a β es obtenida mediante la función $\Delta : [0, g] \rightarrow \tilde{S} = S \times [-0,5, 0,5)$ definida como:

$$\Delta : [0, g] \rightarrow S \times [-0,5, 0,5)$$

$$\Delta(\beta) = (s_j, \alpha); j = [\beta], \alpha = j - \beta$$

donde s_i es la etiqueta con índice más cercano a β , $[\beta]$ entero más cercano a β y α es el valor de la traslación simbólica.

Se debe tener en cuenta que Δ es biyectiva y $\Delta^{-1} : \tilde{S} \rightarrow [0, g]$ se define mediante:

$$\Delta^{-1}(s_i, \alpha) = i + \alpha$$

De esta forma la 2-tupla en \tilde{S} se identifica con el valor numérico en el intervalo $[0, g]$. La conversión de un término lingüístico en un valor 2-tupla lingüístico consiste en añadir un valor 0 como traslación simbólica:

$$s_i \in S \implies (s_i, 0) \in \tilde{S}$$

Junto a este modelo de representación de información lingüística, Herrera y Martínez definieron un modelo

computacional lingüístico basado en las funciones de transformación Δ y Δ^{-1} . En (Herrera and Martínez, 2000) se definieron los operadores de comparación, negación y varios operadores de agregación para 2-tupla que se muestran a continuación.

■ Comparación de 2-Tupla:

La comparación de información lingüística representada mediante 2-tupla se realiza según un orden lexicográfico.

Sean (s_k, α_1) y (s_l, α_2) dos 2-tupla, cada una representando una cantidad de información, entonces:

- si $k < l$ entonces $(s_k, \alpha_1) \prec (s_l, \alpha_2)$
- si $k = l$ entonces
 - si $\alpha_1 = \alpha_2$ entonces $(s_k, \alpha_1) = (s_l, \alpha_2)$;
 - si $\alpha_1 < \alpha_2$ entonces $(s_k, \alpha_1) \prec (s_l, \alpha_2)$;
 - si $\alpha_1 > \alpha_2$ entonces $(s_k, \alpha_1) \succ (s_l, \alpha_2)$.

■ Operador Negación:

$$\text{Neg}(s_i) = s_j \text{ tal que } j = g - i. \quad (1.1)$$

■ Agregación de 2-tupla:

Como se ha indicado, la agregación consiste en obtener un valor colectivo que exprese la información de un conjunto de valores marginales y el resultado debe ser consistente con la representación de los valores de entrada, por tanto, el resultado de la agregación de 2-tupla debe ser una 2-tupla.

En el contexto de la agregación de información, las funciones Δ^{-1} y Δ transforman valores numéricos en 2-tupla y viceversa sin pérdida de información por lo que los operadores de agregación numéricos tradicionales pueden extenderse a 2-tupla de forma intuitiva y fácil.

En la literatura se pueden encontrar diferentes operadores de agregación que obtienen un valor lingüístico 2-tupla a partir de un conjunto de valoraciones lingüísticas en 2-tupla (Liu et al., 2013, Merigó et al., 2010, Peláez and Doña, 2003, Wan, 2013, Wei, 2011, Wei and Zhao, 2012, Yang, 2013, Zhang et al., 2011). A continuación se muestran los que son de mayor interés para esta investigación.

Media Geométrica:

Sea $x = \{(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)\}$ un conjunto de 2-tupla y el vector numérico de la 2-tupla (s_j, a_j)

($j = 1, 2, \dots, n$), la media geométrica en 2-tupla se calcula como (Xu and Huang, 2008):

$$TGA(x) = \Delta \left(\left(\prod_{j=1}^n \beta_j \right)^{\frac{1}{n}} \right), \beta_j = \Delta^{-1}(s_j, a_j) \quad (1.2)$$

Media Aritmética:

Sea $x = \{(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)\}$ un conjunto de 2-tupla, la media aritmética en 2-tupla se calcula como (Herrera and Martínez, 2000):

$$\bar{X}(x) = \Delta \left(\frac{1}{n} \sum_{j=1}^n \beta_j \right), \beta_j = \Delta^{-1}(s_j, a_j) \quad (1.3)$$

Media Geométrica Ponderada:

Sea $x = \{(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)\}$ un conjunto de 2-tupla y $W = (w_1, \dots, w_m)$ el vector numérico de la 2-tupla (s_j, a_j) ($j = 1, 2, \dots, n$), $w_j \in [0, 1]$ y $\sum_{j=1}^n w_j = 1$ la media geométrica ponderada en 2-tupla se calcula como (Xu and Huang, 2008):

$$TWGa_w(x) = \Delta \left(\prod_{j=1}^n \beta_j^{w_j} \right), \beta_j = \Delta^{-1}(s_j, a_j) \quad (1.4)$$

Media Aritmética Ponderada:

Sea $x = \{(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)\}$ un conjunto de 2-tupla y $W = (w_1, \dots, w_m)$ el vector numérico de la 2-tupla (s_j, a_j) ($j = 1, 2, \dots, n$), $w_j \in [0, 1]$ y $\sum_{j=1}^n w_j = 1$ la media aritmética ponderada en 2-tupla se calcula como (Herrera and Martínez, 2000):

$$\bar{X}^w(x) = \Delta \left(\frac{\sum_{j=1}^n \Delta^{-1}(s_j, a_j) w_j}{\sum_{j=1}^n w_j} \right) = \Delta \left(\frac{\sum_{j=1}^n \Delta^{-1} \beta_j \cdot w_j}{\sum_{j=1}^n w_j} \right), \quad (1.5)$$

Media Ponderada Ordenada:

Sea $x = \{(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)\}$ un conjunto de 2-tupla y $W = (w_1, \dots, w_m)$ el vector numérico de la 2-tupla (s_j, a_j) ($j = 1, 2, \dots, n$), $w_j \in [0, 1]$ y $\sum_{j=1}^n w_j = 1$ la media aritmética ponderada ordenada en 2-tupla se calcula como (Merigó et al., 2010):

$$L2TOWA_w(x) = \Delta \left(\sum_{j=1}^n (\beta_j \cdot w_j) \right) \quad (1.6)$$

Con β_j el j -ésimo mayor valor de $\Delta^{-1}(s_j, a_j)$.

1.5. Esquema de resolución de un Problema de Toma de Decisiones Lingüístico basado en el Modelo Computacional 2-Tupla Lingüística

El uso de la información lingüística en la Toma de Decisiones modifica el esquema 1.2 mediante la introducción de dos nuevos pasos, como se muestra en la Figura 1.6. Seguido del esquema se explican las dos nuevas fases.

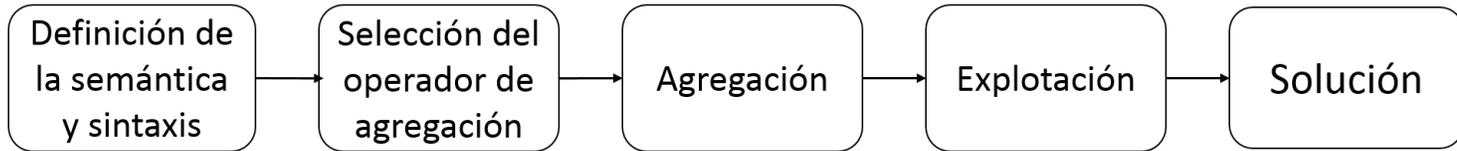


Figura 1.6: Proceso de resolución de un problema de Toma de Decisiones Lingüístico.

Descripción de las fases:

- **La elección del conjunto lingüístico y su semántica:** Establece el dominio de expresión lingüística en la que los expertos ofrecen sus evaluaciones lingüísticas sobre las alternativas de acuerdo con sus conocimientos.
- **La elección del operador de agregación de información lingüística:** Se elige un operador de agregación adecuado para la agregación de las evaluaciones lingüísticas. La idoneidad del operador depende de cada problema de decisión individual.

Hasta el momento, se han revisado, los pasos fundamentales en la solución de un problema de Toma de Decisiones Figura 1.2, cómo se modifican en el caso de un problema de Toma de Decisiones Lingüística Figura 1.6 y las fases fundamentales de un procesos de Toma de Decisiones Figura 1.3. La extensión de estos enfoques para la solución de un problema de Toma de Decisiones Lingüístico basado en el modelo computacional 2-tupla lingüística, se muestra en la Figura 1.7:

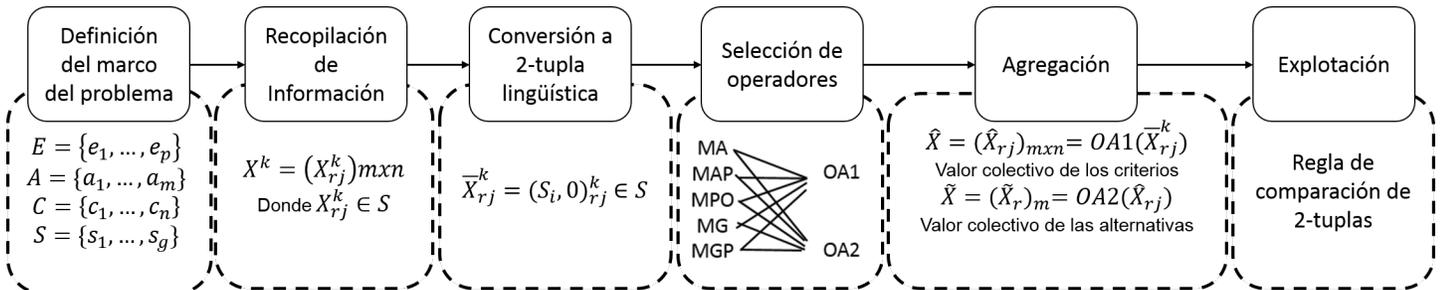


Figura 1.7: Esquema de resolución de un problema de Toma de Decisiones Lingüístico basado en el modelo computacional 2-tupla lingüística

En la definición del marco del problema se generan todos los conjuntos de entrada del problema, la creación de estos no está sujeta a un orden específico.

- $E = \{e_1, \dots, e_p\}$ es el conjunto de expertos que evalúan
- $A = \{a_1, \dots, a_m\}$ alternativas de solución de acuerdo a,
- $C = \{c_1, \dots, c_n\}$ que es el conjunto de criterios, mediante el uso de
- $S = \{s_1, \dots, s_g\}$ un conjunto de términos lingüísticos de granularidad impar.

En la fase de recopilación de información los expertos realizan sus valoraciones sobre los criterios, para cada una de las alternativas. X_{rj}^k representa la valoración lingüística emitida en S del experto k para la alternativa r sobre el criterio j .

En la fase de conversión a 2-tupla lingüística se transforman todas las valoraciones proporcionadas por los expertos. La conversión de un término lingüístico en un valor 2-tupla lingüístico consiste en añadir un valor 0 como traslación simbólica:

$$s_i \in S \implies (s_i, 0) \in \tilde{S}$$

En la selección de operadores se elige el o los operadores de agregación necesarios y más afines con las características del problema. Un operador de los mencionados en la sección 1.4 será OA1 que se utilizará para calcular el valor colectivo de preferencias de los criterios y el OA2 permitirá calcular el valor colectivo de preferencias de las alternativas. Así si el problema es multiexperto con un solo criterio se utilizará el OA1, pero si es multicriterio pero participa un único experto se utilizará el OA2.

Luego se utilizan los operadores seleccionados para agregar las valoraciones emitidas por cada experto, para llegar a un valor de preferencia general para cada alternativa en la fase de agregación.

En la fase explotación se realiza una comparación de las 2-tuplas obtenidas para obtener conjuntos ordenados de alternativas como parte de la solución del problema.

Realizar operaciones de este tipo de forma manual, implicaría principalmente un alto costo temporal no solo por el análisis y procesamiento del conjunto de datos, sino por la complejidad de los cálculos y demás acciones que incluye el proceso como por ejemplo la transformación, manipulación y retransformación de la información. Estos elementos entre otros indican la necesidad del uso de Sistemas de Apoyo a la Toma de Decisiones, que asistan eficientemente la resolución de un problema de Toma de Decisiones Lingüístico basado en el modelo computacional 2-tupla lingüística.

1.6. Sistema de Apoyo a la Toma de Decisiones

El dominio de la definición de los problemas de toma de decisión, ayuda a comprender el significado y objetivo de los Sistemas de Apoyo a la Toma de Decisiones. Se utilizan como ejemplo de partida los conceptos mencionados en algunas bibliografías consultadas.

Para Irina Cendrero y Cristina Colero “son sistemas de información basados en computadora los cuales combinan modelos y datos para intentar resolver problemas no estructurados utilizando una interfaz amigable para el usuario” (Cendrero, 2015).

En (Finlay, 1994) se define un Sistema de Apoyo a la Toma de Decisiones ampliamente como un sistema basado en una computadora que ayuda en el proceso de toma de decisiones.

Para (Keen et al., 1978) los Sistemas de Apoyo a la Toma de Decisiones son un soporte de gestión para los decisores, basado en una computadora que se ocupan de problemas semi-estructurados.

Para (Sprague and Carlson, 1982), los Sistemas de Apoyo a la Toma de Decisiones son sistemas interactivos basados en un ordenador, que ayudan a los decisores, utilizan datos y modelos para resolver problemas no estructurados.

Un sistema de apoyo a la toma de decisiones es un “sistema de información basado en un computador interactivo, flexible y adaptable, especialmente desarrollado para apoyar la solución de un problema de gestión no estructurado para mejorar la toma de decisiones. Utiliza datos, proporciona una interfaz amigable y permite la toma de decisiones en el propio análisis de la situación” (Turban et al., 2005).

Una vez analizados los conceptos de diferentes expertos en el tema y analizar sus puntos comunes se asume la última definición, ya que incluye y complementa los elementos aportados por el resto de los autores. La investigación de estas definiciones aclaró además que los sistemas se pueden clasificar en dependencia de la función que cumplen. A continuación se muestran las cuatro clasificaciones más comunes, junto a una breve explicación de su utilidad (Camacho et al., 2014, Liang and Piza, 2015).

- Sistema de Soporte a la Toma de Decisiones (DSS): Apoyan la Toma de Decisiones mediante la generación y evaluación sistemática de diferentes alternativas o escenarios de decisión.
- Sistema para la Toma de Decisiones en Grupo (GDSS): Su objetivo es lograr la participación de un grupo de personas durante la Toma de Decisiones en ambientes de anonimato y consenso.
- Sistemas de Información para Ejecutivos (EIS): Destinados a apoyar el proceso de Toma de Decisiones de los altos ejecutivos de una organización.

- Sistemas Expertos de Soporte a la Toma de Decisiones (EDSS): Permiten cargar bases de conocimiento que se integran por una serie de reglas de sentido común para que diferentes usuarios las consulten.

La investigación se enfoca en la primera clasificación, al buscar como propuesta de solución el conjunto de alternativas ordenadas de manera descendente comenzando por la mejor. Esta característica se resalta en la enciclopedia en línea “Reference for Bussinnes” al mencionar que los DSS generalmente están diseñados para asistir a la Toma de Decisiones y no intentan tomar la decisión por sí mismos, más bien presentan la información de una manera que es propicia para realizar este proceso de manera informada y eficaz.

A pesar de los avances en el campo de los Sistemas de Apoyo a la Toma de Decisiones y de que el modelo computacional 2-tupla lingüística ha ganado gran aceptación por su interpretabilidad y flexibilidad para adaptarse a diferentes tipos de problema, hasta donde se conoce, solo se ha desarrollado un Sistema de Apoyo a la Toma de Decisión Lingüística basado en 2-tupla lingüística.

1.7. Análisis de Sistemas de Apoyo a la Toma de Decisiones que utilizan el Modelo 2-Tupla Lingüística para el tratamiento de información lingüística

FLINTSTONES es una suite de herramientas que tiene como objetivo la solución de problemas de Toma de Decisión Lingüística definidos en marcos complejos, hace uso del modelo lingüístico 2-tupla y sus extensiones. Está desarrollada como una aplicación Eclipse Rich Client Platform (Eclipse RCP) bajo una arquitectura basada en componentes con el objetivo de resolver problemas comunes en el desarrollo del software como son la reutilización, el mantenimiento, la ampliación y/o la modificación.

Al emplear una arquitectura basada en componentes es necesario asegurar que los diferentes componentes que conforman la aplicación pueden interoperar. Para ello se establece un modelo que fija el estándar a seguir en el desarrollo. Este caso, utiliza el modelo Open Services Gateway Initiative (OSGi), al ser éste el empleado por Eclipse RCP.

Para su desarrollo se implementaron varios componentes OSGi, los cuales pueden ser agrupados en cuatro tipos básicos: interfaz, soporte a los procesos de Toma de Decisiones (bibliotecas), operadores de agregación (operadores) y métodos de resolución lingüísticos que implementan las diferentes extensiones 2-tupla (métodos). Para la implementación de los diversos componentes, se emplearon diferentes tecnologías, mostradas a continuación ([Estrella et al., 2014](#)).

- Eclipse RCP: plataforma de desarrollo de aplicaciones de escritorio orientadas a componentes.
- Java: lenguaje de programación orientado a objetos multiplataforma.

- jExcelApi: biblioteca para la generación y lectura de hojas de cálculo Excel.
- jFreeChart: biblioteca para la generación de gráficas.

FLINTSTONES resuelve problemas de decisión multiatributo, es decir, aquellos en los que las alternativas están caracterizadas por un conjunto de elementos denominados atributos, los cuales tienen asignado un peso, y donde se requiere la asignación de valoraciones individuales para cada atributo de cada una de las alternativas posibles.

La suite permite resolver Problemas de Toma de Decisiones en grupo donde participan múltiples expertos que aportan sus valoraciones sobre las alternativas. Los métodos de resolución implementados están basados en el modelo aditivo (WSM, Weighted Sum Model), lo que permite a la suite incorporar otros métodos de resolución alternativos como los basados en distancias a la solución ideal (Estrella et al., 2014).

El tratamiento de problemas multicriterio y mutiexperto es un valor agregado que presenta esta aplicación, elemento que se convierte en deficiencia en el momento que alcanza su máximo valor de cuatro expertos y cuatro criterios para la solución de los problemas.

Esta herramienta como proyecto inicial fue una buena alternativa que hizo posible un mejor acercamiento a un sistema que agrupara gran parte de las opciones a utilizar para dar solución a un Problema de Toma de Decisiones, pero la evolución teórica y tecnológica ha demostrado que su uso ha disminuido debido a las limitaciones que presenta, principalmente porque no contiene los nuevos operadores y variantes que han surgido con el tiempo.

Es por ello que la propuesta de solución está enfocada a resolver deficiencias como la cantidad de expertos, alternativas y criterios a tratar, e incluir la posibilidad de añadir operadores de forma sencilla en el futuro para enriquecer el contenido del sistema. Además el uso de los operadores de agregación del modelo 2-tupla lingüística permitirá tener más opciones para resolver Problemas de Toma de Decisiones.

Por los motivos expuestos anteriormente se decide desarrollar un sistema Web basado en tecnologías libres, que permita el intercambio de conocimientos entre expertos de distintas ubicaciones geográficas.

1.8. Conclusiones parciales

El uso de información lingüística para modelar información vaga, incierta o imprecisa propia en los procesos cognitivos, es adecuado tanto desde el punto de vista de tratamiento de la incertidumbre, como de la comprensión de los resultados.

El modelo 2-tupla se ha mostrado exitoso en la resolución de problemas de toma de decisión lingüística. Permite realizar procesos de computación con palabras y cumple su regla básica de representar la información

de los expertos y los resultados de forma lingüística. Además preserva la base del Enfoque Lingüístico Difuso, debido a que mantiene la representación lingüística (semántica y sintaxis) de sus resultados.

A pesar de las ventajas del modelo, hasta donde se conoce, no existen numerosas herramientas basadas en 2-tupla. La más importante de ellas, FLINTSTONES, está enfocada en problemas específicos y presenta limitaciones para modelar otras necesidades de los decisores mediante el uso de diferentes operadores de agregación y expertos que puedan acceder a la aplicación desde diferentes ubicaciones para participar en la solución de un mismo problema.

Estos elementos sugieren la necesidad de un Sistema de Apoyo a la Toma de Decisiones basado en el modelo computacional 2-tupla lingüística.

Capítulo 2

Descripción del sistema XABAL 2 TUPLE SOLVE

El presente capítulo muestra la base de partida para la elaboración del sistema XABAL 2 TUPLE SOLVE. Se analizan las tecnologías y herramientas a emplear en la elaboración del sistema que dará solución a la situación planteada. Contiene los principales artefactos a generar en cada etapa, según indica la metodología seleccionada. Para este caso se encuentra la modelación del dominio y sus principales conceptos. Se incluye el levantamiento de requisitos, agrupados en funcionales y no funcionales con los que debe cumplir el sistema, así como los casos de uso y la descripción detallada de cada uno de ellos.

2.1. Herramientas y Tecnologías

Una vez definida la propuesta de solución al problema planteado se hace necesario seleccionar las herramientas y tecnologías con las que será desarrollada. En el presente epígrafe se justifica la elección de las tecnologías y herramientas necesarias para el desarrollo de esta tarea. La solución estará desarrollada con herramientas y tecnologías libres, y sustentada sobre tecnología Web.

2.1.1. Metodología de desarrollo de software

El proceso de desarrollo de software se torna una tarea ardua en la mayoría de las ocasiones. Para lograr obtener un producto con la calidad requerida es necesario tener una guía que imponga cierta disciplina y funcione como un hilo conductor en el proceso de desarrollo. Con este objetivo han sido creadas las metodologías de desarrollo de software, las cuales proporcionan las guías para poder conocer todo el camino a recorrer desde antes de empezar la implementación, con lo cual se asegura la calidad del producto final, así como el cumplimiento en la entrega del mismo en un tiempo estipulado.

Entre las metodologías existentes se define AUP (Agile Unified Process) como la más adecuada para el desarrollo, es una versión simplificada de RUP (Rational Unified Process) que utiliza técnicas y conceptos de este. Se basa en la gestión de riesgos y aboga porque aquellos componentes con alto riesgo tengan más prioridad que los demás y sean desarrollados en etapas tempranas del proyecto. Desarrolla prototipos ejecutables durante la fase de elaboración del producto, demuestra la validez de la arquitectura para los requisitos claves del producto y determina los riesgos técnicos. Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados. El proceso AUP establece un modelo más simple que el que aparece en RUP por lo que reúne en una única disciplina las disciplinas de Modelado de Negocio, Requisitos y Análisis y Diseño. El resto de disciplinas (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las restantes de RUP (Pablo, 2012).

Se eligió esta metodología por ser muy apropiada para guiar proyectos de una complejidad y volumen no muy altos y que necesiten una rápida implementación, los cuales son los aspectos fundamentales a tener en cuenta para el desarrollo del producto a obtener. AUP proporciona un desarrollo del software más rápido y eficiente, con una generación de artefactos media que satisface los requerimientos para la construcción del sistema a la vez que permite un ahorro de tiempo considerable.

Esta metodología cuenta con las siguientes fases ([Pablo, 2012](#)):

1. Concepción: Identificación del alcance y dimensión del proyecto, propuesta de la arquitectura y del presupuesto del cliente.
2. Elaboración: Confirmación de la idoneidad de la arquitectura.
3. Construcción: Desarrollo incremental del sistema, según las prioridades funcionales de los implicados.
4. Transición: Validación y despliegue del sistema.

Para el desarrollo de las fases anteriores las disciplinas de AUP a cumplir son las siguientes ([Pablo, 2012](#)):

- Modelo: Entender el negocio de la organización, el dominio del problema que aborda el proyecto y definir una solución viable.
- Aplicación: Transformar el modelo en código ejecutable y realizar un nivel básico de las pruebas.
- Prueba: Realizar una evaluación objetiva para garantizar la calidad. Esto incluye encontrar defectos, validar que el sistema automatizado funciona según lo previsto y verificar que se cumplan los requisitos.
- Despliegue: Realizar un plan para la presentación del sistema y ejecutarlo para hacer que el sistema se encuentre a disposición de los usuarios finales.
- Gestión de Configuración: Realizar la gestión de acceso a artefactos de su proyecto. Esto incluye no sólo el seguimiento de las versiones del artefacto en el tiempo, sino también el control y la gestión de cambios para ellos.
- Gestión del Proyecto: Dirigir las actividades que se llevan a cabo en el proyecto. Esto incluye la gestión de los riesgos, la dirección de personas (la asignación de tareas, el seguimiento de los progresos, entre otros), y coordinar con las personas para garantizar que se entrega a tiempo y dentro del presupuesto.
- Entorno: Apoyar el resto de los esfuerzos por garantizar que el proceso adecuado, la orientación (normas y directrices), y herramientas (hardware, software, entre otros) están disponibles para el equipo en el momento que ellos lo necesiten.

2.1.2. Lenguaje de modelado

Además de lo esencial que resulta utilizar una metodología de software como hilo conductor de todo el ciclo de vida del sistema, también es importante tener en cuenta que en un proceso de desarrollo de software es necesario contar con algún elemento que describa el aspecto y la conducta del producto, estos elementos son llamados lenguajes de modelado.

UML (Unified Modeling Language) es uno de los lenguajes de modelado de gran utilidad para el desarrollo, pues ofrece un modo estándar de visualizar, especificar, construir y documentar los artefactos de un sistema. Su objetivo es lograr una aplicación informática robusta, flexible y escalable (Jacobson et al., 2000).

Se utiliza la versión 2.0 de UML para la realización de los entregables que propone la metodología seleccionada. Entre los artefactos obligatorios a entregar según la metodología se encuentran el modelo del dominio y modelo de casos de uso para la definición y detalle de los requisitos funcionales. Además se realizan el modelo de diseño, despliegue, objetos y datos con el fin de documentar evidencias que pueden ayudar en la comprensión y desarrollo de futuras versiones. Finalmente se presenta también un modelo de diagrama de componentes de implementación como parte del modelo de implementación. Para la presente investigación se decide utilizar UML 2.0 no solo por las características antes planteadas, sino también por ser el lenguaje de modelado más común en las investigaciones y proyectos realizados por la Universidad.

2.1.3. Herramienta de modelado

Con el objetivo de apoyar y automatizar la metodología de software y el lenguaje de modelado se emplean las herramientas CASE (Computer Aided Software Engineering), las cuales son sistemas que facilitan el diseño y la documentación de las actividades del proceso de desarrollo de un software.

Visual Paradigm¹ V8.0, se considera una de las herramientas CASE más adecuada para trabajar en software libre, esta propiedad es básicamente, la que constituyó un hecho determinante en su selección para ser utilizada en el proceso de modelado del sistema XABAL 2 TUPLE SOLVE. Posee una interfaz amigable y fácil de utilizar. Tiene disponibilidad en múltiples plataformas como Windows, Linux, Unix y en múltiples versiones. Proporciona una plataforma de modelado colaborativo para el trabajo en equipo, los miembros pueden ver y editar el mismo proyecto o esquema, incluso de forma simultánea. Todos los cambios se almacenan en el servidor de Visual Paradigm en función de revisión. Uno de los factores claves para determinar su selección es que soporta todo el ciclo de desarrollo del software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

¹<http://www.visual-paradigm.com>

2.1.4. Lenguaje de programación

PHP² V5.5, Lenguaje Procesador de Hipertextos (del inglés Hipertext Preprocesor), es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML (HiperText Markup Language) resultante de la ejecución del código PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Una de sus grandes cualidades es su versatilidad al momento de escribir el código, su sencillez en la sintaxis e inclusive su seguridad; es gratuito y fácil de aprender. Posee una gran variedad de funciones que pueden ser utilizadas para mejorar el rendimiento de los programas y es un lenguaje utilizado con frecuencia en la Web. Estos elementos denotan que el lenguaje PHP permite lograr la realización de un sistema que cumpla los objetivos propuestos, para garantizar la agilidad y compresión de la etapa de desarrollo.

2.1.5. Entorno de desarrollo integrado

PhpStorm³ V8.0 es un IDE (Integrated Developed Enviroment) de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación PHP como lo indica su nombre. Este IDE posee una excelente integración con marcos de trabajo como Symfony 2, permite la ejecución de códigos de la consola en su mismo entorno y reconoce la sintaxis de varios lenguajes, entre ellos, HTML, CSS (Cascading Style Sheets), JavaScript⁴, PHP, Twig⁵, entre otros.

Dado que JetBrains Company brinda la posibilidad de adquirir una licencia de código abierto para proyectos no comerciales de código abierto la cual es válida por un año se decidió utilizar este IDE en la presente investigación.

2.1.6. Marco de trabajo

Symfony V2.5 es un marco de trabajo de PHP basado en la arquitectura Modelo-Vista-Controlador, que permite en teoría desarrollar aplicaciones de manera más rápida, estructurada y comprensible. Symfony separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias

²<http://php.net>

³<https://www.jetbrains.com/phpstorm/>

⁴Lenguaje de programación interpretado por un navegador Web.

⁵Motor de plantillas para PHP

herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, lo que le permite al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Lo positivo de todas estas ventajas es que no se debe realizar nuevamente algunos procesos cada vez que se crea una nueva aplicación Web (Potencier, 2015). Esta idea es una apuesta inteligente porque se reutilizan conceptos y desarrollos exitosos de terceros, integrados como librerías para ser utilizados por los desarrolladores.

Un ejemplo concreto se evidencia en la capacidad de integrarse plenamente con uno de los marcos de trabajo ORM más importantes dentro de los existentes para PHP llamado Doctrine. Este ORM (Object Relational Mapping) es el encargado de la comunicación con la base de datos, brinda un control casi total de los datos sin importar si se utiliza MySQL, PostgreSQL, SQL Server, entre otros gestores porque la mayoría de las sentencias SQL (Structured Query Language) no son generadas por el programador sino por el mismo Doctrine. Por las posibilidades de ventajas y comodidades para el desarrollo de un sistema, se elige Symfony 2 como marco de trabajo más conveniente.

2.1.7. Sistema gestor de bases de datos

Para almacenar la información que se manipula en un gran porcentaje de aplicaciones, se utilizan los SGBD (Sistema Gestor de Bases de Datos), programas que garantizan la integridad y seguridad de la información y que sirven como intermediarios entre las aplicaciones y los datos.

El gestor de base de datos elegido fue PostgreSQL V9.3 pues tiene soporte nativo para los lenguajes más populares, ejemplo PHP, esto es fundamental porque es el lenguaje de programación que se utiliza para el desarrollo de esta aplicación. Además es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Berkeley Software Distribution) y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y sus últimas versiones no tienen nada que envidiarle a otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto ni el funcionamiento del sistema⁶.

Además de ser altamente configurable, permite personalizaciones de acuerdo al escenario donde será utilizado. Este gestor es multiplataforma, lo que facilita el despliegue de la aplicación. Por las características presentadas anteriormente se decidió seleccionar PostgreSQL como sistema gestor de base de datos.

⁶<http://www.postgresql.org.es/>

2.1.8. Intercambio con el servidor

AJAX es el acrónimo inglés para Asynchronous JavaScript and XML (Extensible Markup Language) (JavaScript y XML asíncrono). Es una técnica de desarrollo Web que genera aplicaciones Web interactivas al combinar:

- Document Object Model (DOM) para visualizar dinámicamente e interactuar con la información presentada.
- XML, XSLT (Extensible Stylesheet Language Transformation) para intercambiar y manipular datos.
- CSS para definir el aspecto (look and feel) del documento.
- JSON (JavaScript Object Notation) y JSON-RPC⁷ pueden ser alternativas a XML/XSLT
- XMLHttpRequest para recuperar datos de forma asincrónica.
- JavaScript como nexo de unión de todas estas tecnologías.

AJAX es un patrón de diseño que propone un nuevo modelo de interacción Web que utiliza las tecnologías anteriores (Eguiluz, 2008, Lázaro, 2002).

2.1.9. Servidor Web

Apache Server⁸ V2.4 es uno de los servidores Web más populares a nivel internacional, se caracteriza generalmente por su gran robustez, flexibilidad, estabilidad y eficiencia. Para el desarrollo y posterior uso del sistema, Apache Server es el servidor Web seleccionado por las múltiples ventajas que ofrece, es modular, de carácter libre, compatible con múltiples sistemas operativos, como Linux y Windows, y es muy popular, por lo que es más fácil conseguir ayuda y soporte. Además, este servidor permite la publicación de documentos PHP de la misma forma que se hace en Internet. Su condición de software libre hace que pueda adaptarse a diferentes entornos y necesidades, no necesita grandes recursos para funcionar y está respaldado por una comunidad de desarrollo amplia. Las características analizadas y el lenguaje de programación Web escogido para la programación del lado del servidor sugieren utilizar Apache Server, para el despliegue del sistema.

2.1.10. Bibliotecas

- **jQuery**

jQuery⁹ V1.10.2 es una biblioteca JavaScript rápida, pequeña y rica en funciones. Hace recorridos del DOM y manipula sus elementos, maneja eventos, realiza animaciones, y Ajax es mucho más simple

⁷Llamada a un procedimiento remoto codificado en JSON

⁸<http://www.apache.org>

⁹<http://jquery.com>

con una API (Application Programming Interface) fácil de usar que funciona a través de múltiples navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript. Ofrece una infraestructura con la que se tendrá mayor facilidad para la creación de aplicaciones complejas del lado del cliente y facilita la creación de interfaces de usuario, efectos dinámicos y aplicaciones que utilizan Ajax.

■ jQuery UI

jQuery UI¹⁰ V1.11.4 es una biblioteca Javascript de componentes para el marco de trabajo jQuery que le añade un conjunto de plug-ins, widgets y efectos visuales para la creación de aplicaciones Web. Cada componente o módulo se desarrolla de acuerdo a la filosofía de jQuery, esto implica la edición de los objetos del DOM . Esta biblioteca añade características como:

1. Menús en acordeón
2. Barras de progreso
3. Botones y casillas de verificación
4. Cuadros de diálogo que se superponen
5. Mecanismo para Autocompletar
6. Mecanismo de pestañas para facilitar la visualización de la página
7. Iconos personalizados
8. Controles deslizantes
9. Calendario

Por las características antes mencionadas se selecciona jQueryUI, para el desarrollo del sistema. En el análisis se consideró también utilizar una identidad marcaria de la universidad lo cual reafirmó la decisión tomada anteriormente.

■ Bootstrap

Twitter Bootstrap¹¹ V2.3 es una colección de herramientas de software libre para la creación de sitios y aplicaciones Web. Contiene plantillas de diseño basadas en HTML y CSS con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript. Bootstrap brinda un útil diseño en Malla, en el cual se puede distribuir el contenido de

¹⁰<https://jqueryui.com>

¹¹<https://www.imaginanet.com/blog/html5-frameworks-html5-boilerplate-y-twitter-bootstrap.html>

la aplicación Web. Este sistema en Malla brinda una total flexibilidad, no importa el dispositivo que se use para visualizar la aplicación o sitio Web, Bootstrap logrará que el entorno de la aplicación se ajuste al dispositivo utilizado. El uso de una identidad marcaría propia de la universidad construida en gran parte con Bootstrap y las características descritas determinaron la selección de esta herramienta.

- **Highcharts**

Highcharts¹² V3.0.9 es una biblioteca de gráficos escrita en JavaScript puro que ofrece una manera fácil de agregar gráficos interactivos a sitios Web o aplicaciones Web. Highcharts actualmente soporta gráficos como: línea, spline, área, área de spline, columna, barras, circulares, de dispersión, medidores angulares ,rango de área de spline, rango de columnas, burbuja, diagrama de caja, barras de error, embudo, cascada y algunos tipos de gráficos polares. Es una herramienta libre de pago que brinda gran calidad en las respuestas, estas características inclinaron la balanza hacia esta en el proceso de selección.

- **PhantomJS**

PhantomJS es un WebKit¹³ de secuencias de comandos con un API de JavaScript. Tiene soporte rápido y nativo para varios estándares Web: manipulación del DOM, selectores CSS, JSON, Canvas¹⁴ y SVG (Scalable Vector Graphics). Permite capturar contenidos de la Web mediante programación, incluye SVG, Canvas y crea capturas de pantalla de sitios Web con vista previa en miniatura. En la presente investigación resulta necesario utilizar este WebKit para exportar los gráficos de la librería Highcharts en el lado del servidor como imagen.

- **Wkhtmltopdf**

Wkhtmltopdf es un pequeño programa en C¹⁵ que convierte HTML y CSS a PDF (Portable Document Format). Al ejecutarse por separado del servidor Web, hace una conversión más rápida. Esta conversión se realiza desde las líneas de comando y no se hace necesario un monitor o alguna forma de visualización de la página que se desea convertir, solo la dirección de esta. Se plantea utilizar este WebKit porque se dispone del Bundle¹⁶ para Symfony2 KNPSnappyBundle el cual funciona como wrapper para wkhtmltopdf que permite de forma sencilla generar un documento PDF.

¹²<http://www.highcharts.com>

¹³Plataforma de aplicaciones que funciona como base para el navegador Web

¹⁴Elemento HTML que permite la generación de gráficos mediante el uso de JavaScript

¹⁵Lenguaje de programación imperativa

¹⁶En Symfony2 es un conjunto estructurado de archivos que se encuentran en un directorio y que implementan una sola característica

2.2. Modelo de dominio

Para (Jacobson et al., 2000) un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los elementos que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Todo ello se representa a través de clases relacionadas, mediante el lenguaje UML, con el objetivo de tener una mejor comprensión de la estructura y dinámica de la organización, los problemas actuales dentro de esta, e identificar las mejoras potenciales.

Para representar los conceptos más significativos en el dominio del problema se realizará una modelación del dominio y se procederá a explicar cada uno de los conceptos que forman parte del mismo, porque no se tienen bien definidos los procesos del negocio, para realizar otras tareas asociadas al área de modelación del negocio.

2.2.1. Diagrama de clases del dominio

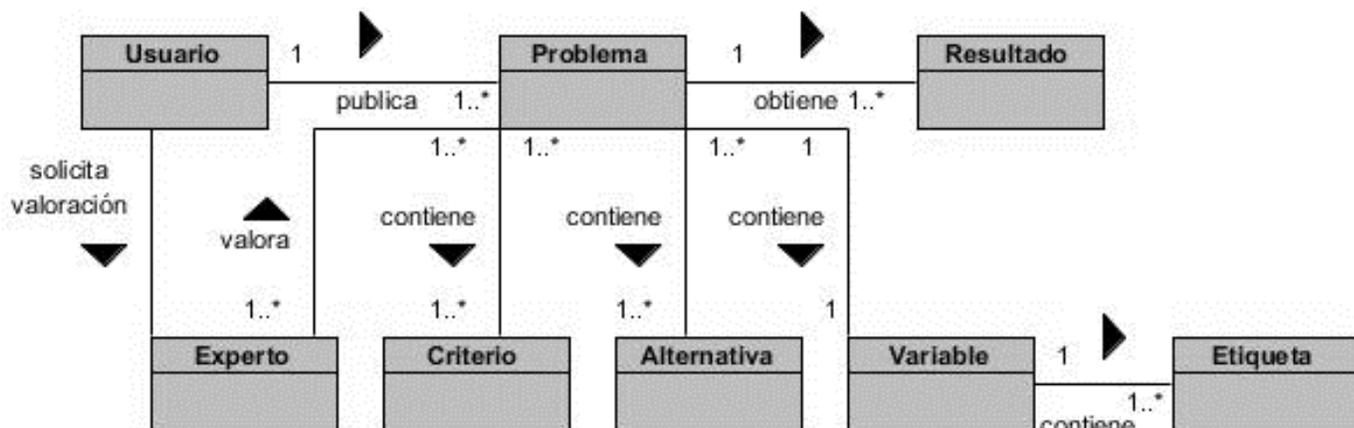


Figura 2.1: Diagrama de clases del dominio

2.2.2. Descripción del modelo de dominio

El usuario publica un problema en busca del conjunto de posibles resultados. El problema estará compuesto por un grupo de alternativas y criterios, sobre las cuales los expertos facilitan su valoración mediante el uso de las etiquetas asociadas a la variable lingüística utilizada, para poder conformar ideas mejor fundamentadas, y brindar un nivel mayor de calidad en sus respuestas. Estas valoraciones se realizarán luego de que el usuario (creador del problema) solicite la opinión de los expertos que dominen este tema.

2.2.3. Definición de las clases del modelo de dominio

- **Usuario:** Persona que accede al sistema con el objetivo de darle solución a un problema.
- **Problema:** Situación de análisis de la cual se espera una solución.

- **Resultado:** Es la solución o respuesta de un problema.
- **Experto:** Persona calificada para dar opinión sobre el tema que esté en análisis.
- **Criterio:** Representa una parte del conjunto analizado para caracterizar una alternativa.
- **Alternativa:** Representa un elemento que forma parte del conjunto analizado que puede ser seleccionado como camino a seguir.
- **Variable:** Representa un conjunto finito de etiquetas lingüísticas.
- **Término lingüístico:** Valor que puede tomar una variable.

2.3. Descripción de la propuesta de solución

Para resolver un problema de Toma de Decisiones con la ayuda de la propuesta de solución se deben seguir los pasos del proceso analizado en la Figura 1.7.

Antes de iniciar la primera fase se requiere que un usuario se registre en el sistema, para poder asociarle los problemas que genere, además podrá realizar valoraciones en otros, en caso de tener la categoría de experto. Pasado este punto se define el marco del problema, donde se generan los conjuntos correspondientes a esta fase. Los problemas podrán tener dos categorías, simple o distribuido, su principal diferencia es que en los distribuidos se solicitará la valoración a expertos reales, mientras que en los simples se podrán simular los expertos.

Luego de tener definidos los conjuntos de expertos, criterios, alternativas y términos lingüísticos se inicia la fase de recopilación de la información, donde los expertos realizarán las valoraciones de cada alternativa según los criterios definidos. Las valoraciones se efectúan mediante el uso del conjunto de términos lingüísticos.

Al concluir el período de valoraciones el usuario que generó el problema puede seleccionar los operadores de agregación que considere correctos para alcanzar los resultados deseados. Los resultados de realizar operaciones de agregación tendrán que ser comparados para poder definir cuál de las alternativas es mejor. Todos estos pasos conforman la fase de valoración de alternativas.

En la etapa final del proceso, llamada explotación, se podrá ver el resultado de la comparación de las alternativas, donde el sistema muestra, según las valoraciones, el mejor camino a seguir. Esta propuesta puede ser aceptada o rechazada por el usuario.

2.4. Requisitos del sistema

Los requisitos del sistema son condiciones que el cliente determina y refina en ocasiones, con la opinión del equipo de trabajo, con el objetivo de resolver un problema que tenga de la vida real. Los mismos pueden ser clasificados en funcionales y no funcionales según indica la ingeniería de software. Para el caso de la presente investigación los requisitos fueron obtenidos mediante un acuerdo entre el cliente y los desarrolladores.

2.4.1. Requisitos funcionales

Los requisitos funcionales (RF) son condiciones o capacidades que debe cumplir el sistema, son acuerdos entre el cliente y los desarrolladores sobre lo que debe o no debe hacer el sistema (Jacobson et al., 2000). Representan en su totalidad las funcionalidades que estarán presentes en el sistema y servirán de guía al proceso de desarrollo. A continuación se muestran los RF del sistema XABAL 2 TUPLE SOLVE seguidos de su breve descripción.

RF 1. “Autenticar usuario”. Esta funcionalidad permite al usuario autenticarse en el sistema para acceder a los problemas asociados a él.

RF 2. “Registrar usuario”. Esta funcionalidad permite al usuario registrar sus datos en el sistema.

RF 3. “Editar perfil de usuario”. Esta funcionalidad permite editar los datos del usuario ya registrado.

RF 4. “Crear experto”. Esta funcionalidad permite adicionar un experto.

RF 5. “Editar experto”. Esta funcionalidad permite editar el nombre de un experto.

RF 6. “Mostrar experto”. Esta funcionalidad permite listar los expertos existentes.

RF 7. “Eliminar experto”. Esta funcionalidad permite eliminar un experto.

RF 8. “Crear alternativa”. Esta funcionalidad permite adicionar una alternativa.

RF 9. “Editar alternativa”. Esta funcionalidad permite editar una alternativa.

RF 10. “Mostrar alternativa”. Esta funcionalidad permite listar las alternativas existentes.

RF 11. “Eliminar alternativa”. Esta funcionalidad permite eliminar una alternativa.

RF 12. “Crear Criterio”. Esta funcionalidad permite adicionar un criterio.

RF 13. “Editar Criterio”. Esta funcionalidad permite editar un criterio.

RF 14. “Mostrar Criterio”. Esta funcionalidad permite listar los criterios existentes.

RF 15. “Eliminar Criterio”. Esta funcionalidad permite eliminar un criterio.

RF 16. “Crear etiquetas”. Esta funcionalidad permite adicionar una etiqueta.

RF 17. “Editar etiquetas”. Esta funcionalidad permite editar una etiqueta.

RF 18. “Mostrar etiquetas”. Esta funcionalidad permite listar las etiquetas existentes.

RF 19. “Eliminar etiquetas”. Esta funcionalidad permite eliminar una etiqueta.

RF 20. “Ordenar etiquetas”. Esta funcionalidad permite ordenar las etiquetas existentes.

- RF 21. “Aplicar operadores”. Esta funcionalidad permite aplicar los operadores en los problemas.
- RF 22. “Emitir valoración”. Esta funcionalidad permite a un experto dar su valoración en un problema.
- RF 23. “Crear problema”. Esta funcionalidad permite adicionar un Problema de Toma de Decisiones.
- RF 24. “Editar problema”. Esta funcionalidad permite editar un Problema de Toma de Decisiones creado por el mismo usuario.
- RF 25. “Eliminar problema”. Esta funcionalidad permite eliminar un Problema de Toma de Decisiones creado por el mismo usuario.
- RF 26. “Solicitar valoración”. Esta funcionalidad permite solicitar la valoración de un experto en determinado problema.
- RF 27. “Exportar a PDF”. Esta funcionalidad permite exportar un problema en un documento PDF.

2.4.2. Requisitos no funcionales

Los requisitos no funcionales (RNF) son requisitos que especifican propiedades del sistema, como restricciones del entorno o la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad ([Jacobson et al., 2000](#)). A continuación se definen los siguientes RNF para el caso del sistema analizado:

RNF 1. Usabilidad

En este requisito se propone que los usuarios que accedan al sistema, hagan uso del mismo aunque no tengan vastos conocimientos sobre la rama de la informática.

RNF 2. Software

Se requiere para los servidores las siguientes condiciones:

- Sistema Operativo: Ubuntu 14.10
- Servidor Web: Apache 2
- Lenguaje del lado del servidor: PHP 5.5
- Sistema Gestor de Base de Datos: PostgreSQL 9.3
- PhantomJS: 1.9.0
- Wkhtmltox 0.12.2.1

Se requiere para las estaciones de trabajo cliente las siguientes condiciones:

- Sistema Operativo: GNU/Linux, Windows.

- Un navegador como Mozilla Firefox, Safari u otro que cumpla con los estándares W3C¹⁷.

RNF 3. Hardware

Se requiere para los servidores las siguientes condiciones:

- Debe poseer tarjeta de red.
- Procesador: Intel Core i5 Segunda Generación o superior
- Memoria RAM: 8 GB
- Disco Duro: 80 GB

Se requiere para las estaciones de trabajo cliente las siguientes condiciones:

- Debe poseer tarjeta de red.
- Procesador: micro dual core 2.0 GHz
- Memoria RAM: 1 GB
- Disco Duro: 40 GB

RNF 4. Apariencia o interfaz externa

Se desea que la interfaz externa del producto sea de fácil navegación por el usuario, sencilla y legible, es por ello que se decidió utilizar una de las identidades marcarias de la UCI (Universidad de las Ciencias Informáticas) debido a que estas han sido sometidas a varias pruebas de aceptación, lo que garantiza un mejor cumplimiento de este objetivo. Además para lograr una interfaz de usuario amigable, atractiva y funcional para el usuario final, es necesario tener en cuenta algunos principios de diseño de interfaz de usuario como los que se muestran a continuación:

- Las funcionalidades deberán estar al alcance de un clic y representadas por íconos asociados a la acción que se realiza, de manera que cualquier persona con un mínimo dominio de la informática pueda hacer uso del sistema.
- Garantizar la legibilidad de manera que exista contraste de los colores de los textos con el fondo y el tamaño de la fuente sea lo suficientemente adecuado a la vista del usuario.
- Los mensajes mostrados al usuario deben ser concisos y de fácil comprensión.
- Los menús y etiquetas de botones deben comenzar con la palabra más importante.

¹⁷World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

Se requiere un certificado SSL.

2.5. Modelo de Casos de Uso del Sistema (CUS)

Una vez recopilados los requisitos funcionales del sistema es necesario conformar el Diagrama de Casos de Uso del Sistema (DCUS). Para tener una mejor comprensión y organización, se agrupan los requisitos según su relación en CUS, además de especificar los actores que interactúan con el sistema y utilizan estos requisitos.

2.5.1. Descripción de los actores que interactúan con el sistema.

Un actor es un usuario del sistema, esto incluye usuarios humanos y otros sistemas computacionales. A continuación en la Tabla 2.1, se mencionan los actores que van a interactuar con el sistema a construir y su descripción a partir del rol que ocupan en el sistema.

Tabla 2.1: Descripción de los actores que interactúan con el sistema

Actor	Descripción
Usuario	Representa los usuarios que van a interactuar con el sistema.
Experto	Representa los usuarios que emiten su valoración.

2.5.2. Identificación de los CUS

(Pressman, 2002) define los CUS como un conjunto de escenarios que identifican una línea de utilización para el sistema que va a ser construido y que facilitan una descripción de cómo el sistema se usará. A continuación se identifican en la Tabla 2.2 los CUS que hacen referencia a los requisitos funcionales identificados para desarrollar el sistema.

Tabla 2.2: Casos de uso del sistema

Referencia a requisitos funcionales	Nombre del CUS
RF1	Autenticar Usuario
RF2	Administrar Registro
RF3	Editar Perfil (extendido)
RF4, RF5, RF6, RF7	Gestionar Experto
RF8, RF9, RF10, RF11	Gestionar Alternativa
RF12, RF13, RF14, RF15	Gestionar Criterio
RF16, RF17, RF18, RF19, RF20	Gestionar Etiquetas
RF23, RF24, RF25	Gestionar Problema
RF3	Editar Perfil (extendido)
RF21	Aplicar Operadores
RF22	Emitir Valoración
RF26	Solicitar Valoración (extendido)
RF27	Exportar a PDF

2.5.3. Diagrama de Casos de Uso del Sistema

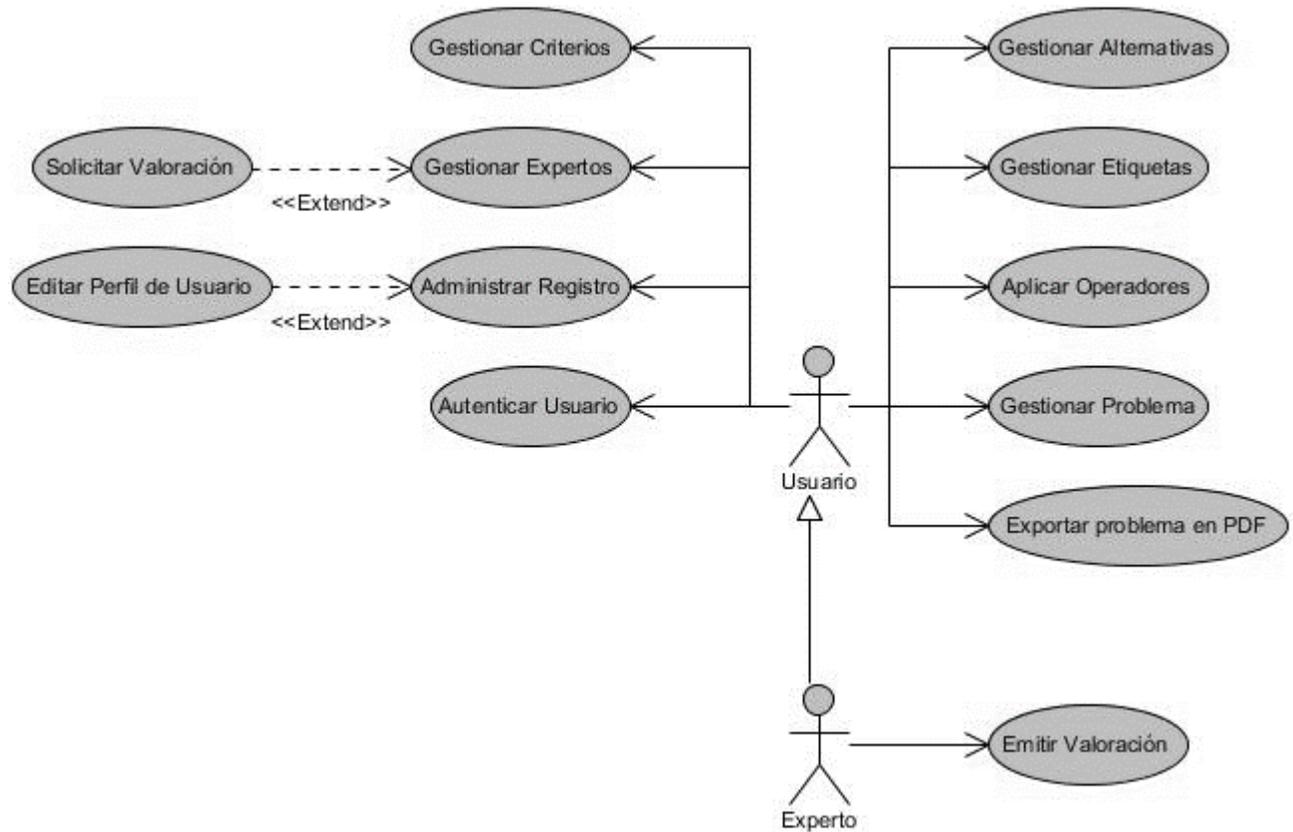


Figura 2.2: Diagrama de Casos de Uso del Sistema

2.5.4. Descripción extendida de los CUS

Debido a la existencia de varios CUS y que las descripciones textuales son muy extensas, se muestra a continuación en la Tabla 2.3 la descripción del Caso de Uso “Aplicar Operador”, las restantes pueden ser consultadas en el documento Anexo titulado “Descripción textual de CUS”.

Tabla 2.3: Descripción del CUS Aplicar Operador

Caso de uso	Aplicar Operador	
Objetivo	Utilizar un operador de agregación del modelo computacional 2-tupla lingüística.	
Actores	Usuario: (Inicia)	
Resumen	Permite utilizar un operador de agregación del modelo computacional 2-tupla lingüística para valorar las alternativas, utiliza las evaluaciones proporcionadas por los expertos.	
Complejidad	Alta	
Prioridad	Alta	
Referencias		
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo básico Aplicar Operadores		
	Actor	Sistema
1	Selecciona la pestaña de Salida dentro del problema.	
2		Muestra la interfaz correspondiente a la ventana de salida de datos.
3	Selecciona los operadores que va a utilizar y presiona el botón "Aceptar".	
4		Muestra la interfaz correspondiente al resultado de aplicar el operador.
Flujos alternos		
Nº Evento 3. El usuario selecciona un operador de peso		
	Actor	Sistema
3.1	Selecciona un operador de peso	
3.2		Muestra los formularios para introducir los pesos asociados al problema para cada operador de peso seleccionado.
3.3	Introduce los valores de peso y presiona el botón "Aceptar".	
3.4		Muestra la interfaz correspondiente al resultado de aplicar el operador.
Nº Evento 3. Existen campos vacíos		
	Actor	Sistema
3.1		Señala en rojo los campos que no deben ser vacíos.
Nº Evento 3. La sumatoria de los pesos asociados no es igual a uno		
	Actor	Sistema
3.1		Muestra un mensaje "Los pesos asociados deben sumar 1"
Relaciones	CU Incluidos CU Extendidos	No procede No procede
Requisitos no funcionales	RNF 1, RNF 2, RNF 3 Y RNF 4	
Asuntos pendientes	No procede	

2.6. Conclusiones parciales

Las herramientas seleccionadas permiten la creación de un sistema que no posea los problemas encontrados en FLINTSTONES. El ambiente Web resuelve principalmente los problemas de gestión de la información y emisión de valoración de los expertos situados en diferentes ubicaciones geográficas.

La biblioteca Highcharts permite una mejor comprensión de los Problemas de Toma de Decisiones al mostrar cómo se representan los conjuntos de términos lingüísticos en su inicio y los desplazamientos que se realizan luego de darles solución. En el modelo 2-tupla lingüística se muestra la representación de su conjunto de términos lingüísticos y sus intervalos de pertenencia basándose en el Enfoque Lingüístico Difuso.

Capítulo 3

Implementación y pruebas del sistema XABAL 2 TUPLE SOLVE

El presente capítulo muestra los artefactos ingenieriles relacionados con el diseño, implementación y validación del sistema a desarrollar. Entre los principales elementos que se muestran se encuentran la arquitectura de software seleccionada y el modelo de diseño, se incluyen los patrones arquitectónicos y de diseño utilizados respectivamente. De igual manera se presentan el modelo de datos, despliegue, implementación y las pruebas realizadas al sistema. Estas evidencias son necesarias para demostrar el correcto funcionamiento de la aplicación desarrollada, a partir de la opinión brindada por el cliente. Se expondrán propuestas de casos de pruebas, junto a sus resultados reales que permitirán mostrar lo antes planteado. Estos objetivos se alcanzarán mediante el uso de varios tipos de técnicas en cada una de las iteraciones.

3.1. Arquitectura de Software

“La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución” (Maier et al., 2001). Así lo define la IEEE (Institute of Electrical and Electronics Engineers), como concepto oficial más común a utilizar por compañías de gran prestigio en el desarrollo de software.

Es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión de este.

Para el desarrollo de la aplicación se decidió seleccionar el marco de trabajo Symfony2, que sigue una arquitectura MVC (Model View Controller) lo que implica que el sistema herede esta arquitectura.

3.1.1. Patrones arquitectónicos

Los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionan un conjunto de subsistemas predefinidos, especifican sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. Un patrón arquitectónico no llega a ser una arquitectura, es más bien un concepto que captura los elementos esenciales de estas. A diferencia de los patrones de diseño estos brindan un nivel mayor de abstracción.

El sistema XABAL 2 TUPLE SOLVE se desarrolló bajo el patrón arquitectónico MVC que es una pro-

puesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

La estrategia de MVC se basa en la separación del código en tres capas diferentes, acotadas por su responsabilidad, llamadas Modelo, Vista y Controlador. Fue creada hace varias décadas incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos marcos de trabajo para el desarrollo Web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones.

3.2. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los CUS centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema, lo que constituye la principal vía de acceso en la actividad de implementación (Jacobson et al., 2000).

3.2.1. Diagrama de paquetes

El diagrama de paquetes es una de las representaciones más comunes en el diseño de una aplicación. Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones (Pressman, 2002). Estos diagramas proporcionan la composición de la jerarquía lógica de un sistema. Los paquetes están normalmente organizados para maximizar la coherencia interna dentro de cada uno y minimizar el acoplamiento externo entre ellos. El siguiente diagrama de paquetes Figura 3.1 muestra la estructura del sistema XABAL 2 TUPLE SOLVE.

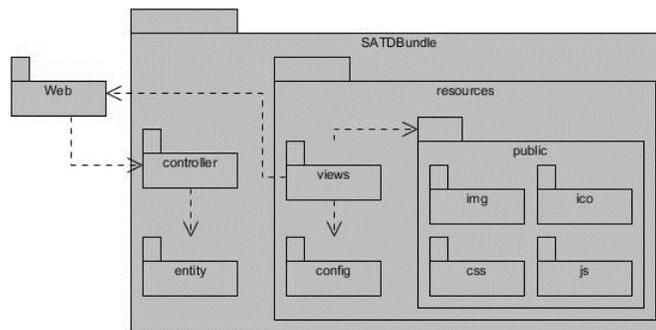


Figura 3.1: Diagrama de paquetes

Breve descripción de los paquetes que componen el sistema:

SATDBundle/controller/ contiene todos los componentes (de extensión php) que consultan los modelos para dar una respuesta.

SATDBundle/entity/ contiene todos los componentes (de extensión php) que mapean las tablas de la base de datos.

SATDBundle/resources/views/ contiene todos los componentes que construyen las vistas.

SATDBundle/resources/config/ contiene todas las direcciones que se manejan en el sistema.

SATDBundle/resources/public/img contiene todas las imágenes que se utilizan en el sistema.

SATDBundle/resources/public/ico contiene todos los iconos que se utilizan en el sistema.

SATDBundle/resources/public/css contiene todos los componentes (de extensión css) que se utilizan en el sistema.

SATDBundle/resources/public/js contiene todos los componentes (de extensión js) que se utilizan en el sistema.

La carpeta `resources` contiene el paquete de vistas (`views`) que conforman la parte visual del sistema, estas hacen uso de los elementos que conforman el paquete `public`. La navegación entre las vistas se realiza con el apoyo de los archivos que se encuentran en `config`. Además se utilizan los paquetes `controller` y `entity` para tratar todas las funcionalidades que brinda el sistema. Todas las peticiones se gestionan a través del paquete `Web` que contiene el controlador frontal de Symfony 2.

3.2.2. Diagrama de Clases del Diseño

Las clases del diseño representan una abstracción de las clases en la implementación de un sistema. El lenguaje para especificar una clase del diseño es el mismo que el lenguaje de programación utilizado. Entre la información general que se puede encontrar en un Diagrama de Clases del Diseño están presentes las clases, asociaciones, atributos, información acerca del tipo de los atributos, interfaces con sus operaciones y constantes, métodos, navegabilidad y dependencias. Los métodos modelados tienen correspondencia directa con los de implementación de clases.

Los Diagramas de Clases del Diseño se emplean en el modelado de las vistas del diseño del sistema para describir las especificaciones de las interfaces y las clases del software, lo que facilita el trabajo de los implementadores. Se propone el siguiente Diagramas de Clases del Diseño para el Caso de Uso “Aplicar Operador” en la Figura 3.2. Los diagramas de los CUS restantes pueden ser consultados en la carpeta “Diagramas de Clases del Diseño” junto a los demás entregables de la carpeta `artefactos`.

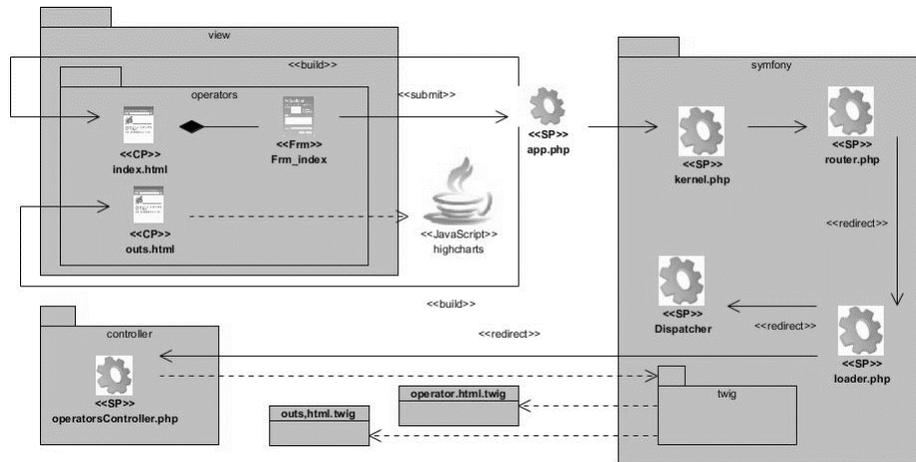


Figura 3.2: Diagrama de Clases del Diseño: Aplicar Operador

3.2.3. Patrones de Diseño

Los patrones de diseño son estructuras o esquemas que pueden aplicarse a varios problemas luego de un proceso de adaptación. Cada uno define cómo ocurre la comunicación entre las clases y objetos del problema. Con su uso se pueden identificar Clases, Instancias, Roles, Colaboraciones y la Distribución de Responsabilidades. Se caracterizan por su reusabilidad, flexibilidad y aplicabilidad a diferentes problemas de diseño en diversas circunstancias. Para el caso del marco de trabajo utilizado se pueden encontrar una serie de patrones de diseño que pertenecen a los conocidos grupos denominados GRASP (General Responsibility Assignment Software Patterns) y GOF (Gang of Four).

En el diseño de la propuesta de solución se aplican los siguientes patrones GRASP, que permiten describir los principios fundamentales de diseño de objetos para la asignación de responsabilidades.

Experto: Este patrón tiene como objetivo principal asignar una responsabilidad determinada a la clase que tenga la mayor cantidad de información para hacer esta tarea. Es la razón anterior la que le da su apellido Experto “en Información”. La propuesta de solución cuenta con la clase Valoraciones donde se puede evidenciar que esta presenta todos los datos necesarios para brindar la información de las entidades relacionadas al problema.

Creador: Se aplica para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto sólo pueda ser creada por el objeto que contiene la información necesaria para ello. En este caso el patrón se refleja en la clase ProblemaRepository, encargada de crear las entidades de Problema, que serán utilizadas posteriormente por AlternativasController, CriterionController, ExpertController y EtiquetaController. Este patrón favorece el bajo acoplamiento entre clases lo que permite una mayor reutilización de esta solución.

Bajo Acoplamiento: El acoplamiento es una medida de la fuerza en que una clase está conectada a otras, que las conoce y recurre a ellas. Para esta solución, se refleja el bajo acoplamiento, en gran parte de las clases del sistema, con el objetivo de que una no dependa de muchas otras, de esta forma, no se afectan las clases por cambios de otros componentes, brinda una mayor facilidad para entenderlos y reutilizarlos.

Alta Cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Se puede afirmar que cada una de las clases del sistema tiene alta cohesión, de manera que estas poseen la característica de tener las responsabilidades estrechamente relacionadas. Esta particularidad evita en cada caso, tener que realizar un trabajo enorme al garantizar un mejor diseño en ocasiones para el resultado global. Este patrón permite que se pueda mejorar la claridad y facilidad en que se entiende el diseño, se simplifique el mantenimiento y existan mejoras de funcionalidad.

Controlador: Para este caso se hace necesario conocer que un evento del sistema es una operación que se realiza en este, generada por un usuario externo. Un controlador, es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Define además el método de su operación. En esta solución, se encuentran ejemplos en las clases `ProblemaController` y `OperatorsController`, esta última es la encargada del comportamiento de los operadores de agregación del modelo y de gestionar todos los eventos que ocurren en el mismo.

Symfony 2 como marco de trabajo, influye en el uso de algunos patrones GOF muchos de estos se implementan indirectamente al interés y vista del usuario. En esta sección se mencionan algunos de los que fueron utilizados para realizar la solución.

El patrón de diseño **Decorator** (Decorador) forma parte de la familia de patrones denominados estructurales. Este tipo de patrones describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Decorator permite modificar, retirar o agregar responsabilidades a un objeto dinámicamente. Para estos ejemplos el término dinámicamente, significa que las funcionalidades se modifican, agregan o retiran durante la ejecución del script o aplicación.

La gran ventaja es que permite extender objetos incluso en situaciones donde la extensión vía herencia no es viable o necesaria. Adicionalmente ayuda a conservar el principio de Abierto/Cerrado, en donde se dicta que cada entidad debe estar abierta a extensión pero cerrada a modificación. Otra ventaja es que las decoraciones evitan la labor de crear clases complejas con mucho código, que en la mayoría de los casos no será evaluado.

El patrón de diseño **Front Controller** (Controlador frontal) es uno de los patrones de diseño de Symfony más conocido. Es una sección de código que atiende todas las solicitudes en la aplicación y devuelve una respuesta al navegador. Además de que este controlador se utiliza para direccionar todas las solicitudes, es el encargado de iniciar el núcleo del sistema lo que le permite decorar el mismo con características adicionales.

3.4. Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. De igual manera describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros (Jacobson et al., 2000).

Un componente es una parte física de un sistema (módulo, base de datos, programa ejecutable), además se puede definir un componente como la materialización de una o más clases.

El diagrama de componentes permite concebir el diseño según los bloques principales y ayuda al equipo de desarrollo a entender un diseño existente y crear uno nuevo. Al establecer el sistema como una colección de componentes con interfaces proporcionadas y necesarias bien definidas se garantiza la correcta separación entre los componentes. A su vez, se facilita la comprensión de los cambios al modificar los requisitos. A continuación se visualiza en la Figura 3.4 la propuesta de diagrama de implementación para el sistema XABAL 2 TUPLE SOLVE.

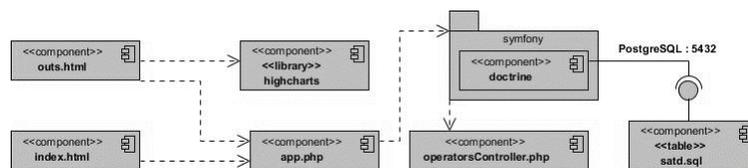


Figura 3.4: Diagrama de Implementación: Aplicar Operador

En el diagrama anterior se puede observar la relación entre los componentes que están presentes en el CUS Aplicar Operador donde `index.html` indica la interfaz gráfica para la selección de operadores. Luego de representar el marco del problema se utiliza `outs.html` para mostrar los resultados a través del controlador `operatorController` y la librería `Highcharts` para visualizarlos gráficamente. Estos componentes se relacionan con las clases internas de `Symfony` mediante el controlador frontal `app.php` lo que permitirá a su vez acceder al modelo con la utilización de `doctrine`.

3.5. Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos (Jacobson et al., 2000), permite modelar la disposición física o topología de un sistema. Muestra el hardware usado y las conexiones físicas presentes, además de los componentes instalados en el hardware y las relaciones entre estos. Sus principales elementos son los nodos, las conexiones y los elementos de anotación.

Nodos: Es un elemento físico que existe en tiempo de ejecución, representa un recurso computacional que, por lo general, dispone de algo de memoria y con frecuencia, capacidad de procesamiento. Un conjunto

de componentes puede residir en un nodo y puede también migrar de un nodo a otro.

Conexiones: El tipo de comunicación es representado por un estereotipo que identifica el protocolo de comunicación o el tipo de red usado.

Elementos de anotación: Los elementos de anotación son las partes explicativas de los modelos UML. Son comentarios que se pueden aplicar para describir, clasificar y hacer observaciones sobre cualquier elemento de un modelo. El tipo principal de anotación es la nota que simplemente es un símbolo para mostrar restricciones y comentarios junto a un elemento o un conjunto de elementos.

A continuación se presenta en la Figura 3.5 la propuesta de despliegue del sistema XABAL 2 TUPLE SOLVE. Su estructura está compuesta por tres nodos físicos que se comunicarán entre sí mediante el uso de dos protocolos. Para la relación del nodo cliente y servidor de aplicaciones se utilizará el acceso por HTTPS (Hypertext Transfer Protocol Secure), mientras que la relación entre los dos servidores se realizará mediante una conexión TCP/IP (Transfer Control Protocol/ Internet Protocol).

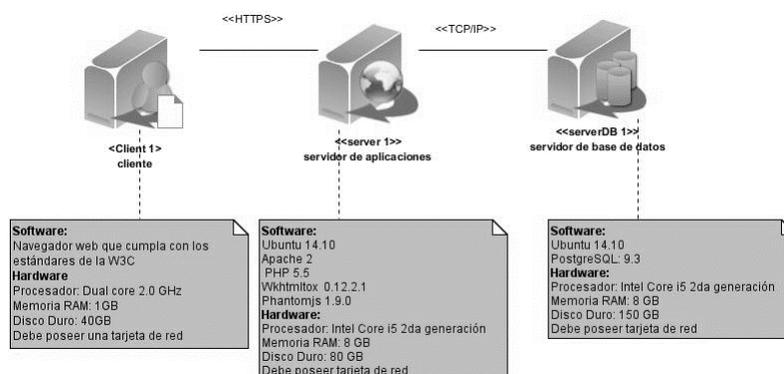


Figura 3.5: Diagrama de despliegue

3.6. Algoritmos implementados

A continuación se muestran algunos de los principales algoritmos implementados que hacen referencia a los operadores de agregación que se estudiaron en la investigación:

```
private function mediaAritmeticaExperts(array $criterios, Problema $problema, $agop_c, array $w = null)
{
    $experts=array();
    foreach ($criterios as $key => $criterio) {
        $experts[$key] = round($this->Ave($criterio, $w)*100)/100;
    }
    .....
}
```

Figura 3.6: Código del algoritmo media aritmética.

```

private function mediaGeometricaExperts(array $criterios, $problema, $agop_c, array $w = null)
{
    $experts=array();
    foreach ($criterios as $key => $criterio) {
        $experts[$key] = round($this->product($criterio, $w)*100)/100;
    }
}

```

Figura 3.7: Código del algoritmo media geométrica.

```

private function OWAExperts(array $criterios, $problema, array $w)
{
    $experts=array();
    arsort($w);
    foreach ($criterios as $key => $criterio) {
        arsort($criterio);
        $experts[$key] = round($this->Ave($criterio, $w)*100)/100;
    }
}

```

Figura 3.8: Código del algoritmo media aritmética ponderada.

3.7. Verificación y validación

La verificación y validación (V & V) de software es un proceso que incluye el análisis y prueba de un sistema que esté en desarrollo o finalizado. Aunque para muchos usuarios parezca lo mismo a primera vista estos conceptos tienen total diferencia, es por ello que, para lograr alcanzar una mejor comprensión del presente epígrafe se hace necesario definirlos correctamente.

La verificación de software responde a la pregunta, ¿Se está construyendo el producto correctamente?, pues su función es “comprobar que el software está de acuerdo con su especificación. Se comprueba que el sistema cumple los requerimientos funcionales y no funcionales que se le han especificado” (Sommerville and Galipienso, 2005). Mientras que la validación de software responde a ¿Se está construyendo el producto concreto?, en esta etapa “se debe asegurar que el software cumple las expectativas del cliente. Va más allá de comprobar si el sistema está acorde con su especificación, para probar que el software hace lo que el usuario espera a diferencia de lo que se ha especificado” (Sommerville and Galipienso, 2005).

La teoría indica que la correcta realización de la fase de análisis de requerimientos debería generar una visión que permita comprender y alcanzar lo deseado por el cliente, pero en realidad hay algunos defectos que solo pueden ser vistos luego de concluir el proceso de desarrollo de los sistemas (Sommerville and Galipienso, 2005).

En el proceso de (V & V) se utilizan dos técnicas para el análisis y comprobación de sistema. Las inspecciones de software analizan y comprueban las representaciones del sistema como el documento de requerimientos, los diagramas de diseño y el código fuente del programa. Se aplican a todas las etapas del proceso de desarrollo. Las inspecciones se complementan con algún tipo de análisis automático del texto fuente o de los documentos

asociados, ambas son técnicas de verificación y validación estáticas puesto que no requieren que el sistema se ejecute. Las pruebas del software consisten en contrastar las respuestas de una implementación del software a series de datos de prueba y examinar las respuestas del software y su comportamiento operacional, para comprobar que se desempeñe conforme a lo requerido. Llevar a cabo las pruebas es una técnica dinámica de la verificación y validación que requiere del uso de un prototipo ejecutable del sistema (Sommerville and Galipienso, 2005).

3.8. Pruebas de Software

Las pruebas de software se definen como un conjunto de actividades previamente especificadas, que se aplican a los sistemas o componentes desarrollados. Se archiva o conserva el registro de casos favorables o no, junto a las salidas obtenidas, para luego realizar una comparación de comprobación donde se analicen los resultados esperados con respecto a los registrados.

Entre las pruebas de software que se aplican comúnmente para comprobar los sistemas se encuentran las funcionales o de caja negra, que comprueban que el sistema responda tal y como el cliente desea. En esta investigación se hace uso de las mismas en conjunto con las pruebas de estrés, con el objetivo principal de demostrar cuan eficiente se comporta el sistema analizado en situaciones de alta concurrencia.

3.8.1. Pruebas de caja negra

Las pruebas de comportamiento o mejor conocidas como pruebas de caja negra, se basan en la comprobación del sistema según las especificaciones requeridas por el cliente. “El componente se ve como una “Caja Negra” cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas” (Juristo et al., 2004). Como el análisis de un sistema de tamaño medio o normal puede generar un elevado número de casos de pruebas, se aconseja realizar las mismas con el juego de datos con mayor probabilidad de fallo. Si el caso generado logra encontrar defectos las pruebas se consideran un éxito, estos defectos deberán ser solucionados en la próxima iteración de prueba junto a todos los errores encontrados.

Para crear los casos de prueba de Caja Negra se pueden utilizar varios criterios de entre los cuales destaca el de particiones de equivalencia debido a su gran popularidad entre las investigaciones consultadas. La presente investigación utiliza la variante antes mencionada bajo el concepto brindado por (Juristo et al., 2004) quien define las particiones de equivalencia como “método de prueba de Caja Negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar”.

A continuación en las Tablas 3.1 y 3.2 se presentan las pruebas realizadas al CUS “Aplicar Operador”, las

pruebas aplicadas a los restantes CUS se encuentran en el documento titulado “Diseño de casos de prueba”, debido a su extensión. Esta información se puede consultar en la documentación anexada a la investigación.

Descripción:

Permite utilizar un operador de agregación del modelo computacional 2-tupla lingüística para valorar las alternativas a partir de las evaluaciones proporcionadas por los expertos.

Tabla 3.1: Casos de prueba del CUS Aplicar Operador

Escenario	Descripción	Variable 1	Variable 2	Variable 3	Variable 4	Respuesta del sistema	Flujo central
EC 1.1: Aplicar operador con éxito.	El usuario selecciona los operadores que desea utilizar.	V Media Aritmética	V Media Aritmética	V I	V I	El sistema realiza el cálculo con los operadores seleccionados y muestra el resultado en un gráfico.	Seleccionar la pestaña "Salida" en el área de trabajo. Seleccionar los operadores correspondientes. Insertar los datos correspondientes (si son de peso). Clic en el botón "Aceptar".
EC 1.2: Aplicar operador con campos vacíos.	El usuario no introduce los datos correspondientes y selecciona el botón "Aceptar".	I (vacío)	V Media Aritmética	V I	V (vacío)	El sistema solicita completar los campos vacíos.	Seleccionar la pestaña "Salida" en el área de trabajo. Seleccionar los operadores correspondientes. Insertar los datos correspondientes (si son de peso). Clic en el botón "Aceptar".
EC 1.3: Aplicar operador con conjuntos diferentes de 1.	El usuario introduce el nombre y selecciona el botón "Aceptar".	V Media Aritmética	V Media Aritmética	I 0.7	I 0.9	El sistema emite un mensaje "Todos los pesos debe sumar 1".	Seleccionar la opción "Etiquetas" del menú lateral. Seleccionar la pestaña "Editar Etiquetas". Seleccionar el botón "Editar". Insertar los datos correspondientes. Clic en el botón "Editar".

Tabla 3.2: Variables del CUS Aplicar Operador

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Operador 1	comboBox	No	Permite seleccionar un operador de agregación.
2	Operador 2	comboBox	No	Permite seleccionar un operador de agregación.
3	Conjunto de pesos de expertos	Spinner	No	Permite introducir el peso por cada experto
4	Conjunto de pesos de criterios	Spinner	No	Permite introducir el peso por cada criterio

Resultado de las pruebas de caja negra

La gráfica presentada a continuación Figura 3.9 muestra los resultados obtenidos de acuerdo a la cantidad de no conformidades detectadas al aplicar las pruebas en cada una de las iteraciones realizadas. Entre los errores detectados se encontraron una funcionalidad incorrecta, dos campos sin validaciones y tres errores de redacción en los mensajes. Las no conformidades detectadas fueron corregidas instantáneamente, sin introducirse nuevos errores en el proceso. La realización de una nueva iteración de prueba confirmó la afirmación antes planteada al mostrar resultados positivos en cada caso.

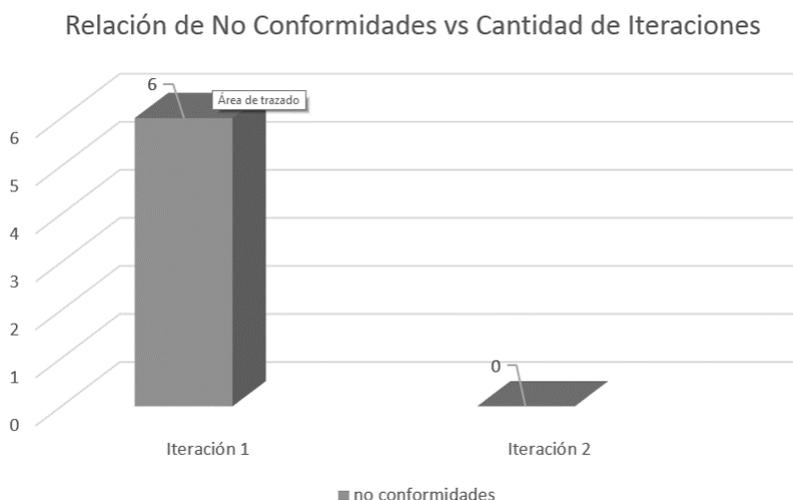


Figura 3.9: Resultados de las pruebas de Caja Negra

3.9. Pruebas de rendimiento

Las pruebas de rendimiento se utilizan para mitigar riesgos relacionados con la continuidad del negocio, potenciar o asegurar la buena imagen corporativa, que podría verse afectada ante el mal funcionamiento de sus aplicaciones. Deben asegurar luego de ser utilizados, el correcto comportamiento de las aplicaciones ante escenarios hostiles o adversos, relativos a manejo intensivo de los recursos o una carga de trabajo inesperada por un alto volumen de acceso concurrente. Validan la escalabilidad, estabilidad y velocidad (tiempos de respuestas) del sistema bajo evaluación, así como los niveles de uso de los recursos para que cumplan con los objetivos de desempeño del sistema.

3.9.1. Pruebas de carga

Constituyen el tipo de pruebas de rendimiento más sencilla. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de esperada peticiones. Un caso de ejemplo puede ser el número esperado de usuarios concurrentes que utilizan una aplicación, con un número

específico de transacciones. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación y los posibles cuellos de botella en la respuesta de determinadas tareas. Su uso puede servir para chequear aspectos como escalabilidad, fiabilidad y explotación de los recursos (Meier et al., 2007).

Resultado de las pruebas de carga

Se ejecutaron las pruebas de carga con la aplicación Apache jMeter en una laptop de 4 GB de RAM y un microprocesador Intel Core I5 de tercera generación para simular un servidor, se configuró como parámetro principal la memoria RAM, para un tamaño de 1 GB. A continuación se muestra la Figura 3.10 que contiene los resultados obtenidos para un total de 100 usuarios con el operador media aritmética como operador de agregación para calcular el valor colectivo de los criterios y para calcular el valor colectivo final de las alternativas.

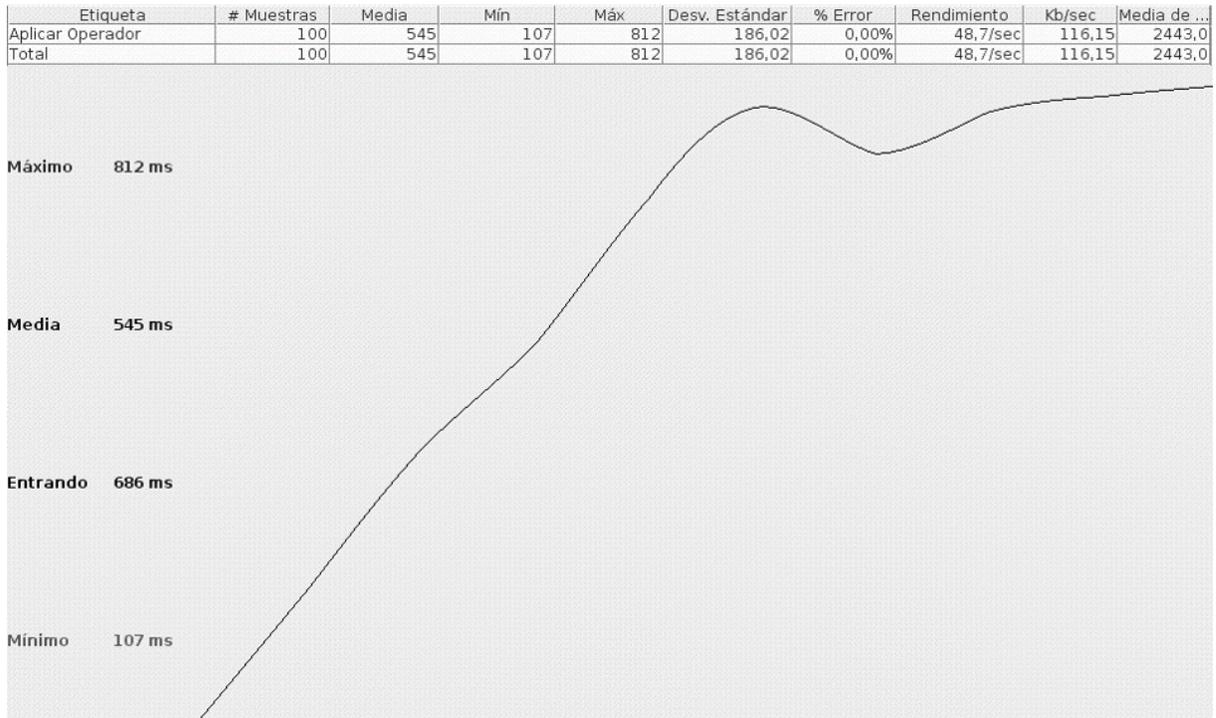


Figura 3.10: Prueba de carga de XABAL 2 TUPLE SOLVE

3.10. Caso de estudio

En este apartado se muestra el funcionamiento XABAL 2 TUPLE SOLVE a través del desarrollo de un ejemplo. El flujo de trabajo está guiado en el esquema de resolución el esquema de resolución de un Problema de Toma de Decisiones Lingüístico basado en el Modelo Computacional 2-Tupla Lingüística presentado en el epígrafe 1.5.

Descripción del Problema: La empresa ETECSA debe comprar nuevos equipos de interconexión para aumentar la disponibilidad del servicio de Internet. Para efectuar la compra se dispone de cuatro proveedores HUAWEI, CISCO, TP-LINK y UBIQUITI, por lo que se hace necesario el análisis de estos según los criterios precio de venta y calidad del producto. Este análisis lo realizaron dos expertos de la Dirección de Redes y Servicios Telemáticos de la empresa.

Definición del marco del Problema: Para el problema el conjunto de expertos estará dado por $E = \{p_1, p_2, p_3, p_4\}$. El conjunto de alternativas a evaluar es $X = \{x_1, x_2, x_3, x_4\}$ (x_1 : HUAWEI, x_2 : CISCO, x_3 : TP-LINK y x_4 : UBIQUITI). El conjunto de criterios está dado por $C = \{c_1, c_2\}$ (c_1 : precio de venta y c_2 : calidad del producto). Los expertos ofrecen sus valoraciones utilizando un término lingüístico del conjunto $S = \{nothing(n), very\ low\ (vl), low\ (l), medium\ (m), high\ (h), very\ high\ (vh), perfect\ (p)\}$. La Figura 3.11 muestra la representación gráfica en el sistema (sintaxis y semántica) del conjunto de términos lingüísticos para este problema. La Figura 3.12 muestra un ejemplo de cómo se gestiona la entrada de datos en el sistema para definir el marco del problema descrito.

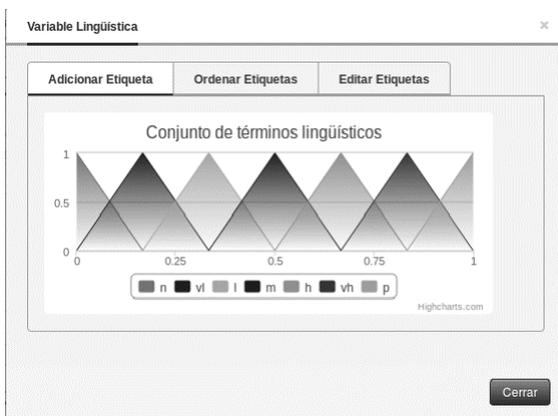


Figura 3.11: Representación gráfica del conjunto de términos lingüísticos de siete términos

Figura 3.12: Gestión de información

Recopilación de información: En esta fase, cada experto introduce en el sistema sus preferencias, utilizando etiquetas lingüísticas para evaluar los diversos criterios por cada una de las alternativas. Las

valoraciones proporcionadas por los expertos se muestran en la Tabla 3.3 y en la Figura 3.13 puede observarse cómo quedarían en el sistema.



Figura 3.13: Valoración de los expertos

Tabla 3.3: Evaluaciones proporcionadas por los expertos.

Criterios	Expertos	Alternativas			
		x1	x2	x3	x4
C1	p1	(vl,0)	(m,0)	(m,0)	(l,0)
C2	p1	(l,0)	(vl,0)	(m,0)	(m,0)
C1	p2	(m,0)	(l,0)	(vl,0)	(h,0)
C2	p2	(l,0)	(m,0)	(l,0)	(m,0)
C1	p3	(h,0)	(vl,0)	(m,0)	(m,0)
C2	p3	(m,0)	(l,0)	(l,0)	(h,0)
C1	p4	(h,0)	(h,0)	(l,0)	(l,0)
C2	p4	(m,0)	(m,0)	(h,0)	(h,0)

Selección del operador de agregación: Luego de tener todas las valoraciones, como parte de la sección de Agregación de Información en XABAL 2 TUPLE SOLVE, donde se seleccionan los operadores que se desean aplicar en el problema. El sistema muestra un total de cinco operadores de agregación en una lista desplegable. Para la solución de este problema se selecciona el operador media aritmética tanto para el cálculo de los valores colectivos de los criterios como de las alternativas.

Agregación y Explotación: Después de realizar la transformación directa de los valores a 2-tupla, estos se agregan utilizando el operador media aritmética. Los resultados, como se muestra en la Figura 3.14, se presentan como una lista ordenada descendientemente de las alternativas evaluadas en el problema. De esta forma el primer elemento, x_4 UBIQUITI, se sugiere como la mejor alternativa bajo el marco del problema definido. La lista es acompañada de un nuevo gráfico donde también aparece la representación gráfica de los resultados. Como puede verse en la Figura 3.15, los resultados para el caso de estudio coinciden con los obtenidos utilizando la suite FLINTSTONES.

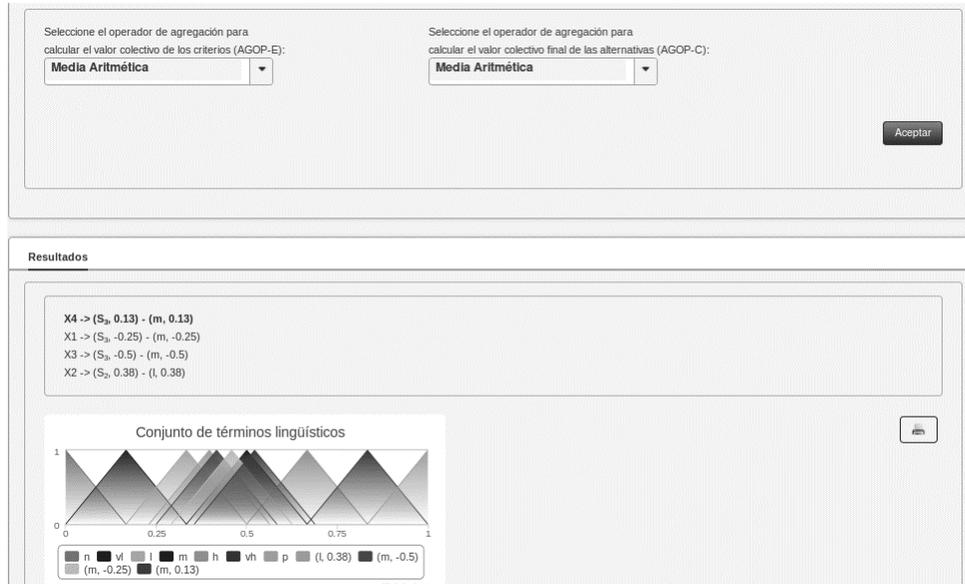


Figura 3.14: Salida del sistema

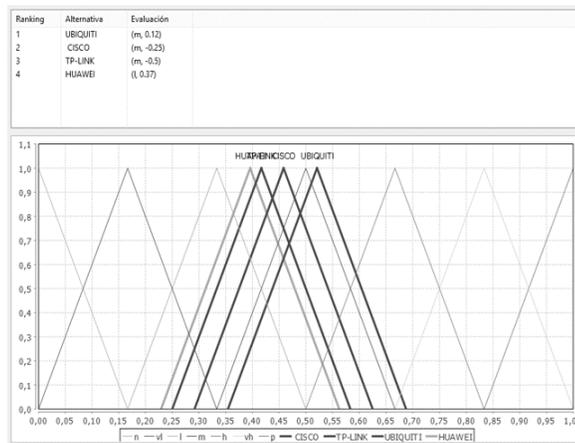


Figura 3.15: Resultado del caso de estudio mediante el uso de FLINTSTONES

3.11. Comparación FLINTSTONES – XABAL 2 TUPLE SOLVE

A continuación se presenta una comparación entre el sistema desarrollado y el estudiado para ilustrar las facilidades de ambas aplicaciones y especialmente los aportes prácticos de XABAL 2 TUPLE SOLVE.

El sistema XABAL 2 TUPLE SOLVE presenta un módulo de registro de usuario que permite diferenciar si un usuario es experto o no, así se garantiza que las personas que realicen valoraciones en un problema sean expertos reales sobre el tema. Además en el caso de tener un problema que requiera confidencialidad se podrá garantizar la misma de esta forma. Se pueden generar problemas simples o distribuidos, donde la principal diferencia se encuentra en que los distribuidos serán valorados por expertos reales, mientras que en los simples se simulará la valoración de los expertos. Sin embargo FLINTSTONES no posee estas características, no realiza gestión de usuarios y tampoco diferencia los problemas lo que puede influir en los resultados de

estos considerablemente.

XABAL 2 TUPLE SOLVE tiene un sistema de notificaciones mediante correo electrónico para gestionar y validar acciones tales como el registro de usuario, cambio de credenciales y solicitud de valoraciones, esta última se aplica solo para usuarios con categoría de experto. Además para el caso de la solicitud de valoraciones el sistema cuenta con una lista de problemas por usuario que también indica si se ha realizado alguna petición en algún problema. FLINTSTONES no presenta ningún sistema de notificaciones que permita a los expertos conocer que se solicitó su valoración en algún problema.

Como complementos principales que brindan un valor agregado a la solución se encuentra la posibilidad de exportar a PDF el problema, antes y después de darle solución. Se presenta un caso de estudio que ejemplifica cómo debe generarse el marco de un problema, además cuenta con la particularidad de ser dinámico ya que se pueden utilizar los operadores que se encuentran en el sistema. Todos los problemas tendrán asociado un campo de descripción que permitirá mejorar la comprensión del problema y los objetivos que se desean alcanzar. La gestión de la información se realiza de forma ágil permitiendo al usuario generar una cantidad determinada de Expertos, Alternativas, Criterios y Términos Lingüísticos, este último con un máximo de 15 términos y siempre con cardinalidad impar debido a las características del modelo. A pesar de que FLINTSTONES permite exportar problemas en Excel, no los muestra con una estructura que permita comprender el objetivo que persigue el problema.

El sistema XABAL 2 TUPLE SOLVE solo resuelve problemas de Toma de Decisiones Lingüísticos utilizando el modelo computacional 2-tupla lingüística, lo que constituye su mayor limitación con respecto a la suite FLINTSTONES, que permite resolver problemas empleando extensiones del modelo 2-tupla lingüística.

En la tabla [3.4](#) se muestra una comparación de estas características en ambos sistemas.

Tabla 3.4: Comparación FLINTSTONES – XABAL 2 TUPLE SOLVE

Características		FLINTSTONES	XABAL 2 TUPLE SOLVE
Marco de Trabajo	Expertos	4	sin restricciones
	Alternativas	4	sin restricciones
	Criterios	4	sin restricciones
	Términos Lingüísticos	sin restricciones	15
Naturaleza del problema		Las valoraciones de los expertos son manejadas por un único usuario (Simple)	Las valoraciones de los expertos son manejadas por un único usuario (Simple). Los expertos proveen sus valoraciones de forma independiente (Distribuidos)
Modelos		2-tupla y extensiones	2-tupla lingüística
		Heterogéneo	
		Multigranular	
		No balanceado	
Otras funcionalidades			Registro de usuario
		Exportar como Excel	Notificación por correo
		Importar datos de Excel	Exportar a PDF
			Ejemplos interactivos
			Descripción del problema

A pesar de que el sistema XABAL 2 TUPLE SOLVE solo permite resolver problemas de Toma de Decisiones Lingüísticos, este se desarrolló siguiendo una estructura que permite incorporar las extensiones del modelo computacional 2-tupla lingüística mediante módulos de mediana complejidad.

3.12. Conclusiones parciales

En la etapa de validación y verificación las pruebas de carga permitieron corroborar la estabilidad del sistema alcanzando una media de respuesta de 545 milisegundos para la media aritmética, operador que es comúnmente utilizado para dar solución a los Problemas de Toma de Decisiones.

Los resultados que se obtienen en el sistema cumplen con la regla de la computación con palabras, porque se obtienen resultados lingüísticos luego de operar sobre un conjunto de términos en un dominio lingüístico.

Conclusiones generales

A partir de los retos planteados en los objetivos, se ha desarrollado un trabajo de investigación que ha permitido llegar a las siguientes conclusiones:

- El estudio de los modelos computacionales: “Modelo basado en el principio de extensión” y “Modelo simbólico” permitió comprobar que en la manipulación de la información, generalmente estas soluciones no muestran los resultados en el mismo dominio lingüístico de expresión inicial, se pueden realizar aproximaciones para llevar los resultados a este dominio, pero este proceso implica la pérdida de información en los resultados alcanzados.

- La utilización del modelo 2-tupla lingüística permitió obtener un Sistema de Apoyo a la Toma de Decisiones que preserva las reglas básicas de la computación con palabras y que muestra el resultado final en el mismo dominio lingüístico del conjunto inicial, lo cual brinda una mejor comprensión de los resultados obtenidos.

- Se desarrolló el Sistema Web XABAL 2 TUPLE SOLVE, considerando múltiples criterios y expertos y con la posibilidad de emplear los operadores de agregación media aritmética, media aritmética ponderada, media geométrica, media geométrica ponderada y media ponderada ordenada lo que permite modelar y resolver problemas de Toma de Decisión Lingüística de diversa naturaleza. Las tecnologías y arquitectura empleadas, facilitan la incorporación de nuevas funcionalidades, con lo que se obtiene una herramienta flexible, escalable, adaptable para todos los entornos.

- En la etapa de validación y verificación se ejecutaron pruebas que permitieron demostrar que los operadores del modelo no implican la necesidad de crear algoritmos complejos que dependan de servidores de grandes prestaciones para realizar las operaciones en entornos Web.

Recomendaciones

1. Integrar nuevos operadores de agregación del modelo computacional 2-tupla lingüística que permitan modelar otras necesidades de los decisores y por tanto enriquecer la solución de los problemas analizados.
2. Flexibilizar y ampliar el marco de trabajo mediante el desarrollo de nuevas funcionalidades que permitan la modelación y resolución de problemas definidos en otros contextos complejos bajo incertidumbre, para los que ya se han definido extensiones del modelo computacional 2-tupla lingüística, entre ellos:
 - Tratamiento de información heterogénea: para operar con información lingüística, numérica, intervalar, entre otros tipos de datos.
 - Tratamiento de información lingüística multigranular: para incluir diferentes conjuntos de términos lingüísticos con diferente granularidad.
 - Tratamiento de escalas no balanceadas: para incluir escalas que no necesariamente estén simétricamente distribuidas.

Bibliografía

- Miguel Badaracco. Sistema tutor inteligente basado en competencias (sti-c). propuesta de arquitectura y diagnóstico. *Universidad de Málaga*, 2013.
- Gabriela Barberis. La toma de decisiones multicriterio eficaces en el ámbito de la pequeña y mediana empresa: El establecimiento de los objetivos y la determinación de los criterios. fac cc. económicas y empresariales. universidad san pablo,[en línea]. 2000.[consultado el 8 de marzo 2013].
- Graciela Elisa Barchini. Métodos i+ d de la informática. *Revista de Informática Educativa y Medios Audio-visuales*, 2(5):16–24, 2005.
- Piero Bonissone and Keith Decker. Selecting uncertainty calculi and granularity: An experiment in trading-off precision and complexity, uncertainty in artificial intelligence. pages 217—247.
- Piero P. Bonissone. A fuzzy sets based linguistic approach: theory and applications. In *Proceedings of the 12th conference on Winter simulation*, pages 99–111. IEEE Press, 1980.
- Gloria Bordogna and Gabriella Pasi. A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation. *Journal of the American Society for Information Science*, 44(2):70, 1993.
- James J. Buckley. The multiple judge, multiple criteria ranking problem: A fuzzy set approach. *Fuzzy sets and Systems*, 13(1):25–37, 1984.
- Francisco Javier Cabrerizo. Nuevos modelos de toma de decisión en grupo con información lingüística difusa. 2008.
- G. Camacho, L. García, E. Rodríguez, and J. Yopez. D.s.s. (sistemas de soporte a la toma de decisiones). 2014.
- Noemí de Castro García et al. Números realistas vs. dos-tuplas. 2013.
- Colero C. Cendero. D.s.s. (sistemas de soporte a la toma de decisiones). 2015.
- Rosanna Degani and Giovanni Bortolan. The problem of linguistic approximation in clinical decision making. *International Journal of Approximate Reasoning*, 2(2):143–162, 1988.
- Miguel Delgado, Francisco Herrera, Enrique Herrera-Viedma, and Luis Martínez. Combining numerical and linguistic information in group decision making. *Information Sciences*, 107(1):177–194, 1998.
- Javier Eguiluz. Introducción a ajax. 2008.

- Macarena Espinilla Estévez. Nuevos modelos de evaluación sensorial con información lingüística. *DEA, Universidad de Jaén. Jaén: sn*, 2009.
- Francisco Javier Estrella, Macarena Espinilla, and Luis Martínez. Flintstones: Una suite para la toma de decisiones lingüísticas basada en 2-tupla lingüísticas y extensiones. In *XVII Congreso español sobre tecnologías y lógica fuzzy*, 2014.
- Paul N. Finlay. Introducing decision support systems. (2), 1994.
- Rosario Garza, Caridad González, and Elvis Salinas. Toma de decisiones empresariales: un enfoque multicriterio multiexperto. *Ingeniería Industrial*, 28(1):7, 2007.
- Rosario Garza, Caridad González, and Elvis Salinas. Aplicación de las técnicas multicriterio multiexpertos dentro del perfil del ingeniero industrial. *Ingeniería Industrial*, 26(1):12–pág, 2010.
- Don Hellriegel, Susan E. Jackson, John W. Slocum, and Pilar Mascaró Sacristán. *Administración: un enfoque basado en competencias*. Thomson learning, 2002.
- Francisco Herrera and Luis Martínez. A 2-tuple fuzzy linguistic representation model for computing with words. *Fuzzy Systems, IEEE Transactions on*, 8(6):746–752, 2000.
- Francisco Herrera and Luis Martínez. The 2-tuple linguistic computational model: advantages of its linguistic description, accuracy and consistency. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(supp01):33–48, 2001.
- Francisco Herrera, E. Herrera-Viedma, and Luis Martínez. A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy sets and systems*, 114(1):43–58, 2000.
- Francisco Herrera, Enrique Herrera-Viedma, and Luis Martínez. A fuzzy linguistic methodology to deal with unbalanced linguistic term sets. *IEEE Transactions on fuzzy Systems*, 16(2):354–370, 2008.
- Francisco Herrera, Sergio Alonso, Francisco Chiclana, and Enrique Herrera-Viedma. Computing with words in decision making: foundations, trends and prospects. *Fuzzy Optimization and Decision Making*, 8(4):337–364, 2009.
- Ivar Jacobson, Grady Booch, and James Rumbaugh. *El proceso unificado de desarrollo de software*, volume 7. Addison Wesley Reading, 2000.
- Natalia Juristo, Ana M. Moreno, and Sira Vegas. Técnicas de evaluación de software. *Visitar http://is.ls.fi.upm.es/udis/docencia/erdsi/Documentacion_Evaluacion_7.pdf*, 2004.

- Peter G.W. Keen, Michael S Scott, et al. Decision support systems; an organizational perspective. Technical report, 1978.
- Rolando Alfredo Hernández León and Zayda Coello González. El paradigma cuantitativo de la investigación científica. *La Habana: Editorial universitaria*, 2002.
- Rolando Alfredo Hernández León and Zayda Coello González. *El paradigma cuantitativo de la investigación científica*. Editorial Universitaria, 2008.
- Jing Liang and David Piza. D.s.s. (sistemas de soporte a la toma de decisiones). 2015.
- Hongbin Liu, Jianfeng Cai, and Luis Martínez. The importance weighted continuous generalized ordered weighted averaging operator and its application to group decision making. *Knowledge-Based Systems*, 48: 24–36, 2013.
- Issi Lázaro. Javascript. 2002.
- Mark W. Maier, David Emery, and Rich Hilliard. Software architecture: introducing ieee standard 1471. *Computer*, 34(4):107–109, 2001.
- Luis Martínez, Jun Liu, Da Ruan, and Jian-Bo Yang. Dealing with heterogeneous information in engineering evaluation processes. *Information Sciences*, 177(7):1533–1542, 2007.
- Luis Martínez, Rosa M. Rodríguez, and Francisco Herrera. The 2-tuple linguistic model: Computing with words in decision making, 2015.
- J. Meier, Carlos Farre, Prashant Bansode, Scott Barber, and Dennis Rea. *Performance testing guidance for web applications: patterns & practices*. Microsoft press, 2007.
- Jerry M. Mendel, Lotfi A. Zadeh, Enric Trillas, Ronald Yager, Jonathan Lawry, Hani Hagrass, and Sergio Guadarrama. What computing with words means to me. *IEEE Computational Intelligence Magazine*, 5(1): 20–26, 2010.
- José M. Merigó, Montserrat Casanovas, and Luis Martínez. Linguistic aggregation operators for linguistic decision making based on the dempster-shafer theory of evidence. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(03):287–304, 2010.
- Torrecilla Pablo. El proceso unificado Ágil: fases y disciplinas. 2012.
- José Ignacio Peláez and Jesús M. Doña. Lama: A linguistic aggregation of majority additive operator. *International Journal of Intelligent Systems*, 18(7):809–820, 2003.

- Roger S. Pressman. *Ingeniería del software un enfoque práctico*, 2002.
- Manuel Quesada. *Modelo de resolución de problemas de toma de decisión multicriterio*. 2015.
- Rosa M. Rodríguez, Luis Martínez, and Francisco Herrera. Toma de decisión lingüística. modelos computacionales simbólicos y su aplicación en el tratamiento de percepciones.
- Rosa M. Rodríguez, Luis Martínez, and Francisco Herrera. Hesitant fuzzy linguistic term sets for decision making. *IEEE Transactions on Fuzzy Systems*, 20(1):109–119, 2012.
- Ian Sommerville and María Isabel Alfonso Galipienso. *Ingeniería del software*. Pearson Educación, 2005.
- Ralph H. Sprague and Eric D. Carlson. *Building effective decision support systems*. Prentice Hall Professional Technical Reference, 1982.
- Richard M. Tong and Piero P. Bonissone. A linguistic approach to decisionmaking with fuzzy sets. *Systems, Man and Cybernetics, IEEE Transactions on*, 10(11):716–723, 1980.
- Vicenç Torra et al. Negation functions based semantics for ordered linguistic labels. *International Journal of Intelligent Systems*, 11(11):975–988, 1996.
- Vicenc Torra. Aggregation of linguistic labels when semantics is based on antonyms. *International Journal of Intelligent Systems*, 16(4):513–524, 2001.
- Efraim Turban, J. Aronson, and Ting-Peng Liang. *Decision Support Systems and Intelligent Systems 7 Edition*. Pearson Prentice Hall, 2005.
- Yeleny Zulueta Véliz. *Modelos de evaluación de la importancia del impacto ambiental en contextos complejos bajo incertidumbre*. Editorial de la Universidad de Granada, 2014.
- Shu-Ping Wan. Some hybrid geometric aggregation operators with 2-tuple linguistic information and their applications to multi-attribute group decision making. *International Journal of Computational Intelligence Systems*, 6(4):750–763, 2013.
- Gui-Wu Wei. A method for multiple attribute group decision making based on the et-wg and et-owg operators with 2-tuple linguistic information. *Expert Systems with Applications*, 37(12):7895–7900, 2010.
- Gui-Wu Wei. Some harmonic aggregation operators with 2-tuple linguistic assessment information and their application to multiple attribute group decision making. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 19(06):977–998, 2011.

- Gui-Wu Wei and Xiaofei Zhao. Some dependent aggregation operators with 2-tuple linguistic information and their application to multiple attribute group decision making. *Expert Systems with Applications*, 39(5): 5881–5886, 2012.
- Yejun Xu and Ling Huang. An approach to group decision making problems based on 2-tuple linguistic aggregation operators. In *2008 ISECS international colloquium on computing, communication, control, and management*, volume 3, pages 73–77. IEEE, 2008.
- Zeshui Xu. A method based on linguistic aggregation operators for group decision making with linguistic preference relations. *Information sciences*, 166(1):19–30, 2004.
- Ronald R. Yager. Concepts, theory, and techniques a new methodology for ordinal multiobjective decisions based on fuzzy sets. *Decision Sciences*, 12(4):589–600, 1981.
- Ronald R. Yager. An approach to ordinal decision making. *International Journal of Approximate Reasoning*, 12(3):237–261, 1995.
- Ronald R. Yager. On the retranslation process in zadeh’s paradigm of computing with words. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(2):1184–1195, 2004.
- Ronald R. Yager and Vladik Kreinovich. Decision making under interval probabilities. 1999.
- Wei Yang. Induced choquet integrals of 2-tuple linguistic information. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 21(02):175–200, 2013.
- Lotfi A. Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.
- Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3):199–249, 1975.
- Lotfi A. Zadeh. Fuzzy logic= computing with words. *Fuzzy Systems, IEEE Transactions on*, 4(2):103–111, 1996a.
- Lotfi A. Zadeh. Nacimiento y evolución de la lógica borrosa, el soft computing y la computación con palabras: un punto de vista personal. *Psicothema*, 8(2):421–429, 1996b.
- Longchang Zhang, Hua Zou, and Fangchun Yang. A dynamic web service composition algorithm based on topsis. *Journal of networks*, 6(9):1296–1304, 2011.

Apéndice A

Características de los Operadores de Agregación

La correcta realización de todo el proceso puede permitir la obtención de mejores resultados aunque es necesario hacer énfasis en las fases 2 y 3, debido a que en estas se selecciona y utiliza, el o los operadores de agregación que se ajusten mejor a la estructura del problema. Al resolver Problemas de Toma de Decisiones con el modelo 2-tupla lingüística es necesario utilizar como mínimo un operador, que se rige por los siguientes criterios de selección:

1. Fuerza axiomática: Un operador se considera más adecuado que otro si está menos limitado por los axiomas que debe satisfacer.
2. Ajuste empírico: Los operadores deben ajustarse a la realidad y al contexto del problema lo que hace más adecuado determinados operadores según el problema que se intente solucionar. La realización de pruebas empíricas, especialmente con valores conflictivos, suele ser ventajosa para seleccionar un operador.
3. Adaptabilidad: Los operadores deben poder adaptarse al contexto cambiante en el que se encuentran, principalmente mediante la parametrización.
4. Eficiencia numérica: El esfuerzo computacional de cálculo necesario es importante a la hora de resolver problemas grandes y puede ser un factor clave para elegir un operador u otro.
5. Compensación y rango de compensación: A medida que mayor sea el grado en que se contrarrestan los argumentos agregados, el operador se adaptará mejor y ofrecerá mejores soluciones.
6. Comportamiento agregado: El número de elementos agregados puede influir en los resultados, por tanto, este hecho debe ser tenido en cuenta al elegir el operador.
7. Nivel de escala requerido: Algunos operadores pueden requerir diferentes niveles de escala para ser admisibles. En igualdad de condiciones se prefiere el que aporte un nivel de escala más bajo.

Un operador de agregación es definido formalmente a partir de sus propiedades matemáticas, como:

Una función $\Psi : [0, 1]^n \rightarrow [0, 1]$ es llamada operador de agregación si cumple las propiedades siguientes:

(Monotonía) $\Psi(x_1, \dots, x_n) \leq \Psi(y_1, \dots, y_n)$ siempre que $x_i \leq y_i$ para todo $i \in \{1, \dots, n\}$,

(Identidad) $\Psi(x) = x$ para todo $x \in [0, 1]$,

(Contorno) $\Psi(0, \dots, 0) = 0$ y $\Psi(1, \dots, 1) = 1$.