Universidad de las Ciencias Informáticas

Facultad 3

Componente para la construcción de resúmenes lingüísticos a partir de los datos de los procesos penales de la Fiscalía General de la República

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

Autor: Pedro Justo Placencia Díaz
Tutor: MSc. Carlos Rafael Rodríguez Rodríguez

La Habana, 2016
Año 58 de la Revolución

Declaración de autoría

Declaro ser autor del presente trabajo de diploma, y le confiero con carácter permanente, el derecho de uso para el desarrollo institucional a la Universidad de las Ciencias Informáticas.

Para que así conste, firmo la presente a los ____ días del mes de julio del año 2016.

Pedro Justo Placencia Díaz

_____
Firma del autor

MSc. Carlos Rafael Rodríguez Rodríguez

_____
Firma del tutor

Dedicatoria y agradecimientos

Dedicatoria

Este trabajo va dedicado a todos mis amigos, a mi familia, a todas aquellas personas que me han apoyado y me han impulsado para seguir adelante. Va dedicado muy especialmente a la persona que más amo en la vida, mi madre.

Agradecimientos

A lo largo de nuestra vida conocemos muchísimas personas, algunas pasan inadvertidas, otras se olvidan con el paso del tiempo y solo unas pocas marcan tu vida a través de ese sentimiento que aflora en la amistad. A mis amigos quiero agradecerles por los los momentos inolvidables, por el apoyo incondicional y por el simple hecho de saber que están ahí siempre que los necesito.

Quiero agradecerle a una persona muy especial en mi vida que llegó cuando menos lo esperaba, y que muy a su forma, me ha brindado su amor y su apoyo. Gracias por iluminar mis días grises.

Y no por último es menos importante, agradecer a mi linda familia, a mis tías y a mis primos con las que he podido contar siempre, a mi hermana y muy especialmente a mi mamá. Gracias mami, por no rendirte nunca a pesar de las circunstancias, gracias por ser una luchadora, gracias por la paciencia, por la confianza, por el amor, por ser la persona que nunca me falla. Enfocado en tí, van todos mis logros.

Resumen

El crecimiento de la información almacenada en bases de datos y la necesidad de explorar la relación entre los datos en busca de conocimiento nuevo, supone un reto de suma importancia. La correcta interpretación de estos datos es una capacidad importante que incide en el proceso de toma de decisiones, lo que es crucial para el futuro de cualquier organización.

En este trabajo se propone un enfoque de sumarización lingüística de datos que permite resumir atributos tanto cualitativos como cuantitativos almacenados en una base de datos. Este enfoque basado en cuantificadores difusos, es utilizado para resumir lingüísticamente un conjunto de atributos relacionados con los procesos penales de la Fiscalía General de la República de Cuba, específicamente de las materias Ordinario y Sumario. La información obtenida una vez aplicado el enfoque, es mostrada mediante resúmenes en un lenguaje natural entendible por el ser humano.

Palabras claves: conocimiento, datos, decisiones, lingüística, sumarización.

Abstract

The growth of information stored in databases and the need to explore the relationship between data for new knowledge is a challenge of paramount importance. The correct interpretation of data is an important capability that affects the decision-making process, which is crucial for the future of any organization.

This paper presents an approach to linguistic summarization of data that allows to summarize both qualitative and quantitative attributes stored in a database. This approach based on fuzzy quantifiers, is used to summarize a set of linguistically related terms of criminal data proceeding from the General Attorney of the Republic of Cuba, specifically from the Ordinary and Summary matters. Information obtained after the linguistic summarization of data application is shown by abstracts in an understandable natural human language.

Key words: data, decision-making, knowledge, linguistic, summarization.

Índice general

Índice de figuras

Índice de tablas

Introducción

El desarrollo de las tecnologías de la información y las comunicaciones ha venido aparejado al crecimiento de volúmenes de datos. En la actualidad, muchas empresas e instituciones producen cantidad de información derivada de los procesos que en ellas se realizan. Esta información, es generalmente almacenada en bases de datos debido a la necesidad de que los datos perduren en el tiempo.

Según estudios realizados por la EMC Corporation, (empresa fabricante de software y sistemas para administración y almacenamiento de información), se estima que entre el año 2015 y 2020 el volumen de información digital almacenada estaría alrededor de los 40 Zettabyte (Gantz y Reinsel, 2012). En la actualidad el análisis de grandes cantidades de información, con el objetivo de obtener conocimiento nuevo, supone un reto de suma importancia. Muchas veces se tiene un gran número de datos y se hace necesario descubrir y conocer la relación entre estos.

La correcta interpretación de estos datos es una capacidad importante que incide en el proceso de toma de decisiones y selección de estrategias, por lo que es crucial para el futuro de la organización. Podría afirmarse que nada es tan valioso como poseer la información correcta en el momento adecuado.

Sin embargo, las personas tienen dificultades al analizar grandes volúmenes de datos que en ocasiones no saben cómo tratar. Actualmente se cuenta con herramientas para acopiar enormes cantidades de información que generalmente se ha almacenado en bases de datos. Este cúmulo de información crea una inminente necesidad y buenas oportunidades para encontrar conocimiento que puede estar implícito en ese conjunto de datos y que no se puede extraer a simple vista.

El descubrimiento de conocimiento en bases de datos (KDD por sus siglas en inglés, Knowledge Discovery in Databases) es un proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia entendibles en los datos (Fayyad et al., 1996). En lo general, KDD es utilizado para resolver problemas relacionados con grandes cantidades de datos (Sánchez et al., 2015).

La sumarización lingüística de datos (LDS por sus siglas en inglés: Linguistic Data Summarization), es considerada un enfoque de descubrimiento de conocimiento para extraer patrones a partir de la información almacenada en bases de datos. LDS permite captar y describir brevemente las tendencias y las características que aparecen en un conjunto de datos. Su uso es especialmente factible pues proporciona resúmenes que no son tan concisos como los estadígrafos media, mediana, varianza, etc. y permite el tratamiento de datos numéricos o no (Yager, 1982).

La sumarización lingüística se entiende como un lenguaje natural, generalmente corto, como una frase, capaz de resumir la esencia de un conjunto de datos numéricos o no, y por lo general demasiado grande

para ser comprendido por el ser humano (Kacprzyk y Zadrożny, 2005).

La Fiscalía General de la República (FGR) emplea para la gestión de su actividad judicial una aplicación informática sobre la web basada en tecnologías libres. En este sistema se gestionan los procesos de las áreas Protección a los Derechos Ciudadanos, Verificaciones Fiscales, Control de la Legalidad en los Establecimientos Penitenciarios y Procesos Penales. En esta última área es donde se concentra entre el 60 y el 70 por ciento de los procesos de la FGR y por tanto es ahí donde se genera la mayor cantidad de datos.

Este volumen de datos no siempre es posible utilizarlo con efectividad para tomar decisiones, debido fundamentalmente a las siguientes limitaciones:

- Generalmente se tienen muchas instancias de cada tipo de trámite compuestos por varios atributos.
- Generalmente existe presencia de atributos numéricos (1, 2, 3) o simbólicos (nombre, provincia) para una misma instancia.

Las dos limitaciones anteriores podrían entenderse como que la información contenida en la base de datos sobre los procesos penales puede llegar a ser no entendida por los usuarios. Esto:

- Dificulta y en casos extremos impide la identificación de relaciones entre los atributos del proceso que no resulten triviales.
- Limita la identificación de tendencias positivas o negativas en cuanto a algún atributo del proceso.
- Demanda de los decisores mayor tiempo para poder detectar e interpretar comportamientos subyacentes.

De la problemática descrita, se identifica como **problema a resolver**: ¿Cómo contribuir al proceso de extracción de conocimiento de forma que favorezca el apoyo a la toma de decisiones sobre los procesos penales de la Fiscalía General de la República de Cuba?

Tomando como **objeto de estudio**, la extracción de conocimiento de bases de datos.

Siendo el **campo de acción**, la sumarización lingüística de datos.

Se define como **objetivo general** desarrollar un componente para la obtención de resúmenes lingüísticos que contribuyan a la toma de decisiones sobre los procesos penales de la Fiscalía General de la República de Cuba.

Para darle solución al objetivo general planteado, se trazaron los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación haciendo uso de métodos empíricos, teóricos y particulares para determinar las principales tendencias en el campo de la Sumarización Lingüística de Datos y su aplicación a la Informática Jurídica.
- Modelar el componente propuesto, mediante la metodología de desarrollo de software seleccionada.

- Implementar en PLR todas las funcionalidades definidas en el componente modelado, mediante las herramientas y tecnologías seleccionadas.
- Validar y verificar el componente propuesto mediante la aplicación de pruebas de software.

Se define como **Idea a defender**:

Si se desarrolla un componente para la obtención de resúmenes lingüísticos, de manera que permita la extracción de conocimiento, se contribuirá al apoyo a la toma de decisiones sobre los procesos penales de la Fiscalía General de la República de Cuba.

Como **posibles resultados**:

- El diseño y modelado de un componente para la sumarización lingüística de los datos de los procesos penales de la Fiscalía General de la República de Cuba.
- La implementación en PLR del componente diseñado.

Métodos de investigación

Para dar cumplimiento al objetivo general y realizar las tareas de investigación, se han combinado un grupo de métodos y procedimientos teóricos de la investigación científica, los cuales son:

Dentro de los métodos teóricos

Analítico-Sintético: el método se emplea para analizar el estado del arte del proceso de sumarización lingüística de datos y de esta forma obtener conocimiento procediendo a sintetizarlo.

Histórico-lógico: el método consiste en realizar una revisión exhaustiva del desarrollo evolutivo del objeto de investigación a lo largo del tiempo con el objetivo de definir las limitaciones actuales de su conocimiento. Este método se utilizó para reconocer los avances teórico-prácticos y problemas que actualmente existen en el ámbito de la Informática Jurídica y la Sumarización Lingüística de Datos.

Inducción-Deducción: son las formas de razonamiento que permiten llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización. Mediante este método se logró definir el enfoque de Sumarizción Lingüística de Datos a utilizar para el desarrollo de la solución propuesta.

Modelación: método que opera en forma práctica o teórica con un objeto, no en su forma directa, sino utilizando cierto sistema intermedio, auxiliar, natural o artificial. Mediante la modelación se crearon abstracciones del sistema propuesto como las tarjetas CRC.

Dentro de los métodos empíricos

Entrevista: implica la recopilación de información mediante una conversación profesional. Los resultados a lograr dependen en gran medida del nivel de comunicación entre el investigador y los participantes en

la misma. Puesta en práctica mayormente con fiscales, para obtener conocimiento acerca de los diferentes términos que se manejan durante el desarrollo de la investigación.

El trabajo está estructurado de la siguiente manera: introducción, desarrollo dividido en tres capítulos, seguido de las conclusiones generales del trabajo, bibliografía consultada y referencias bibliográficas y al final del trabajo los anexos. Seguidamente se describe la distribución de contenidos en cada uno de los tres capítulos.

Capítulo #1: Fundamentación Teórica

En este capítulo se definen los conceptos relacionados con la Informática Jurídica y la Sumarización Lingüística de Datos, además de un estudio del estado del arte enmarcado en el desarrollo de ambos términos en el mundo y en Cuba. Se definen las herramientas y tecnologías a utilizar para el desarrollo de la solución propuesta, además de la metodología para guiar el proceso de desarrollo de software. Al finalizar, las conclusiones parciales del capítulo.

Capítulo #2: Solución Propuesta

En el desarrollo de este capítulo se describe la estructura del componente que se propone como solución. Esto se realiza describiendo las funcionalidades con sus entradas y salidas, además de las actividades de que se realizan dentro de cada funcionalidad. Se desglosan en secciones las fases de la metodología de desarrollo de software Planificación, Diseño y Codificacion con las características específicas de cada una. Al finalizar, las conclusiones parciales del capítulo.

Capítulo #3: Fase de Prueba de la Solución

En este capítulo se describe la fase de pruebas de la metodología de desarrollo de software. Para esto, se calculan los valores de los pesos definidos mediante el AHP, necesarios para la obtención del grado total de validez de los resúmenes, se realizan los casos de prueba al sistema implementado y finalmente se muestran los resultados. Se describe además, la técnica de Iadov para establecer el grado de satisfacción de los usuarios con respecto a la utilidad de los resúmenes lingüísticos, como herramienta de apoyo a la toma de decisiones.

Capítulo 1

Fundamentación Teórica

En el presente capítulo se precisan un conjunto de elementos que conforman la fundamentación teórica de la investigación. Se abordan los temas relacionados a la toma de decisiones y la Informática Jurídica y se exponen los aspectos fundamentales de la sumarización lingüística de datos y el estado actual de estos temas en conjunto. Se definen las herramientas, tecnologías y la metodología por la que se rige el equipo de trabajo para el desarrollo de la aplicación.

1.1. Bases conceptuales

En esta sección se enuncian los principales conceptos y definiciones que constituyen el marco teórico de la investigación, siendo estas las bases principales que le dan soporte.

1.1.1. Toma de decisiones

De acuerdo con diversas investigaciones y enfoques teóricos, la toma de decisiones puede definirse como: *Un proceso amplio que puede incluir tanto la evaluación de las alternativas, el juicio, como la elección de una de ellas* (Labra, 1998). En otras palabras, la toma de decisiones hace referencia a la capacidad cognitiva para elegir; lo que involucra: análisis, categorización, juicios probabilísticos, construcción de alternativas y decisión. La toma de decisiones es un proceso intencional que combina el análisis de la información, la confrontación de alternativas, la valoración de las opciones y, finalmente, la toma de la decisión (Gómez, 2011).

La toma de decisiones es un aspecto importante dentro de cualquier organización. En una era de cambiante tecnología y creciente competencia, pocas organizaciones pueden darse el lujo de basar sus decisiones en reacciones intuitivas y espontáneas, o corazonadas (Vélez Pareja, 2000). El desarrollo de sistemas informáticos ha devenido en una poderosa herramienta de apoyo para la toma de decisiones. Por citar un ejemplo, dentro de la administración empresarial, para cualquier directivo se hace indispensable dotarse de una herramienta que facilite el análisis de información. Una empresa en su día a día genera miles de números y datos que si no son utilizados de forma correcta pueden ocasionar problemas y dificultades en la propia empresa, al no saber como analizarlos (Claramonte, 2012).

1.1.2. Informática Jurídica

La Informática Jurídica está orientada hacia el desarrollo de aplicaciones, software, hardware y demás elementos de la ciencia de la computación para mejorar el quehacer de los juristas, como herramientas de apoyo. Trata más que, de modernizar el derecho mismo, de modernizar la práctica de algunos procedimientos jurídicos (Orozco, 2013).

Tiene como finalidad constituirse en un apoyo para el profesional del Derecho a través del uso de los computadores y las nuevas tecnologías, para de esta forma acceder más eficientemente a la información jurídica de manera más rápida. Pretende evitarle al profesional del Derecho tareas engorrosas como sumergirse en archivos, desplazarse de un sitio a otro para buscar información útil o estar casi esclavizado, en ocasiones, en su oficina para llevar a cabo su labor (Orozco, 2013).

La informática jurídica *busca aplicar al Derecho la lógica y otras técnicas de formalización, con vistas al empleo de los medios electrónicos (...) procura adquirir las técnicas adecuadas de los ordenadores al campo jurídico, incluida la necesaria para la construcción y el empleo de programas específicos tendientes a este fin* (Guibourg et al., 1996).Es la informática como tecnología aplicada al Derecho con el fin de solventar de manera mejor las necesidades de los juristas (Orozco, 2013).

Al igual que sucede en el resto de las disciplinas, la informática se ha convertido en una herramienta jurídica indispensable. La informática jurídica pretende asistir a los profesionales del derecho a desarrollar las competencias, (García Izquierdo, 2005) y se puede englobar en las siguientes tres clasificaciones (Guibourg, 2015).

- Informática Jurídica Documental.
- Informática Jurídica de Gestión.
- Informática Jurídica Decisoria.

Esta última es la rama de la Informática Jurídica que más controversia genera, pues es asumida como la posibilidad de dejar los procesos de decisión en potestad de un ordenador. Se concibe como la opción de que un ordenador haga justicia por sus propios medios (Orozco, 2013).

De acuerdo con las tres clasificaciones anteriores, se han creado sistemas de Informática Jurídica (Tellez, 1996). Estos sistemas se clasifican en:

- Sistema de Informática Jurídica Documental.
- Sistema de Informática Jurídica de Control y Gestión.
- Sistemas de Informática Jurídica Metadocumentaria.

1.1.2.1. Informática Jurídica Documental

La Informática Jurídica Documental: es la parte de la informática jurídica que persigue el almacenamiento de datos (leyes, decretos, resoluciones, fallos judiciales u otros documentos jurídicos, así como referencias acerca de ellos o información bibliográfica) y su clasificación de acuerdo con criterios apropiados para su recuperación rápida y oportuna. Tiende a cumplir por medios electrónicos lo que manualmente se hacía con tomos de legislación, repertorios de jurisprudencia u otras publicaciones de consulta, también dotadas de nomencladores e índices minuciosos (Tellez, 1996).

La primera manifestación de la Informática Jurídica está dada por la creación de grandes bases de datos, y fue bastante bien recibida por los operadores del derecho, que vieron en ella una ayuda valiosa. Hoy en día, el debate sobre métodos de clasificación y canales de búsqueda, que dominaba el pensamiento de los especialistas en las décadas de 1970 y 1980, se ha visto superado por la eficiencia de los buscadores automáticos en el ámbito de Internet (Tellez, 1996).

Un sistema de Informática Jurídica Documental consiste en la creación y recuperación de información jurídica como leyes, doctrina y jurisprudencia. En ésta se trata de crear un banco de datos jurídicos (o corpus jurídico documentario) relacionado con las fuentes del derecho, a efecto de interrogarlo con base en criterios propios acordes a esa información y su relevancia jurídica. La finalidad de un sistema documentario consiste en encontrar lo más rápida y pertinentemente posible la información almacenada. El conjunto de esas informaciones constituye el banco de datos (Tellez, 1996).

1.1.2.2. Informática Jurídica de Control y Gestión

La Informática Jurídica de Gestión busca elaborar nuevos datos a partir de los que se almacenan y presentarlos bajo una nueva forma a fin de cumplir necesidades o funciones jurídicas. Trata de aplicar la informática (y la telemática) a las actividades de gestión, de los escritos de los abogados, documentación de los jueces, operaciones de ministerios públicos y en general, la aplicación de las nuevas tecnologías en las funciones que desempeñan cotidianamente los operadores del derecho (Estrada y Miranda, 2010).

Los sistemas de Informática Jurídica de Control y Gestión abarcan los ámbitos jurídico-administrativo, judicial, registros y despachos de abogados. A través de estos sistemas se pueden obtener datos jurídicos como contratos, certificaciones, mandatos judiciales, entre otros. Tienen como antecedentes el tratamiento de textos jurídicos, mediante el uso de procesadores de la palabra y, por otra parte, las experiencias obtenidas en materia de automatización de registros públicos (Tellez, 1996).

1.1.2.3. Informática Jurídica Decisoria

Es una ciencia que estudia la utilización de aparatos o elementos físicos electrónicos, como la computadora, en el derecho; es decir, la ayuda que este uso presta al desarrollo y aplicación del derecho. En otras palabras, es ver el aspecto instrumental dado a raíz de la informática en el derecho (Joana, 2011).

La informática jurídica de ayuda a la decisión es el tratamiento automatizado de las fuentes del conocimiento jurídico a través de los sistemas de documentación legislativa, jurisprudencial y doctrinal, etcétera.

Se basa en el empleo de técnicas de la IA para la creación de sistemas expertos que ayuden a la toma de decisiones en diferentes planos y niveles (Joana, 2011).

Antonio Anselmo Martino considera que un sistema experto es aquél que, partiendo de ciertas informaciones proporcionadas por un especialista en la materia considerada, pretende resolver problemas que se presentan al interior de un específico *dominio* mediante la simulación de razonamientos que expertos han obtenido por sus conocimientos y experiencias adquiridas (Martino, 1998).

1.1.3. Lógica difusa

La lógica difusa fue investigada por primera vez alrededor de mediados de los años sesenta por el ingeniero Lotfy A. Zadeh en la Universidad de Berkeley (California). En un principio este ingeniero no denominó a esta lógica como lógica borrosa, sino que la llamó principio de incompatibilidad. Describió él este principio como: "Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes".

Se puede definir a este tipo de lógica como una técnica de la inteligencia computacional que ayuda o permite trabajar con información que es imprecisa y no está bien definida. Pertenece a la lógica multivaluada, pero se diferencia de esta, en que nos permite introducir valores intermedios entre la afirmación completa o la negación absoluta (Matías y Vicente, 2008). Este tipo de lógica tiene sus raíces en la teoría de los conjuntos difusos, enunciado por primera vez por Lotfi A. Zadeh (Zadeh, 1996).

1.1.3.1. Conjuntos difusos

La teoría de los conjuntos difusos constituye el punto de partida en el desarrollo de la lógica difusa. Las bases de esta teoría quedan establecidas en el trabajo de Lotfi Zadeh publicado en 1965 en la revista *Information and Control*. En esta publicación se introduce por primera vez de manera formal la definición de un conjunto difuso. Esto da origen a una serie de conceptos, operaciones y medidas que son aplicables a innumerables disciplinas de la ciencia (Pérez y León, 2007).

Un conjunto difuso expresa el grado de pertenencia al conjunto que tiene cada uno de los elementos, dentro del intervalo [0,1] donde (1- pertenencia total al conjunto y 0- no pertenece). El conjunto difuso A en X puede definirse como el conjunto de los pares ordenados

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

donde $\mu_A(x)$ es la función de pertenencia al conjunto difuso, ver figura 1.1.

1.1.4. Funciones de pertenencia

Se define función de pertenencia como aquella aplicación que asocia a cada elemento de un conjunto difuso el grado con que pertenece al valor lingüístico asociado. Los conjuntos difusos son caracterizados

por sus funciones de pertenencia. Un conjunto borroso está determinado por funciones de pertenencia asociadas a él (Cox et al., 1998). Si X es una colección de objetos denotados genéricamente por x, entonces un conjunto borroso A en X se define como un conjunto de pares ordenados:

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

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

1.1.5. Variables lingüísticas

Según (Zadeh, 1974) por una variable lingüística queremos decir una variable cuyos valores son palabras o frases en un lenguaje natural o artificial. Una variable lingüística se caracteriza por una quíntuple $(L >, T(L), U, G, M)$ en la que L es el nombre de la variable; $T(L)$ es el término-conjunto de L, es decir, el conjunto de sus valores lingüísticos; U es un universo de discurso; G es una regla sintáctica que genera los términos en $T(L)$; y M es un regla semántica que asocia a cada valor lingüístico X su significado, $M(X)$, donde $M(X)$ denota un subconjunto borroso de U. El significado de un valor lingüístico X se caracteriza por una función de la compatibilidad, $c : U \rightarrow [0, 1]$, que asocia a cada elemento u, del universo de discurso U, su compatibilidad con X .

El concepto de una variable lingüística proporciona un medio de caracterización aproximada de fenómenos que son demasiado complejos o demasiado mal definidos para ser susceptibles de descripción en términos cuantitativos convencionales. En particular, el tratamiento de la *verdad* como una variable lingüística con valores como la verdad, muy cierto, totalmente cierto, no muy verdadera, falsa, etc., nos lleva a lo que se llama la lógica difusa.

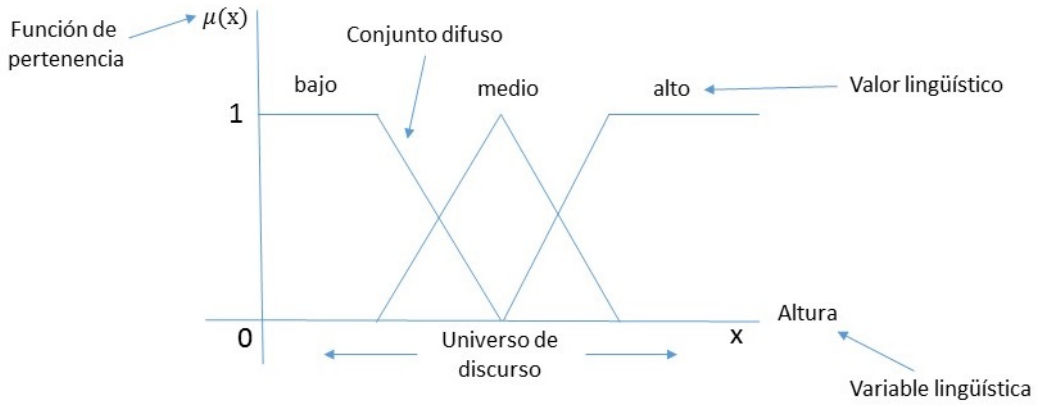


Figura 1.1: *Ejemplo de representación de conjuntos borrosos para la variable altura.*

1.1.6. Sumarización lingüística de datos

La sumarización lingüística de datos (LDS por sus siglas en inglés: Linguistic Data Summarization) es considerada como un tipo minería de datos o un enfoque de descubrimiento de conocimiento para extraer patrones de bases de datos. Se entiende como un lenguaje natural, generalmente corto, como una frase, capaz de resumir la esencia de un conjunto de datos numéricos o no, y por lo general demasiado grande para ser comprendido por el ser humano (Wu y Mendel, 2011).

Existen distintos enfoques para la sumarización lingüística de datos. El más usado es el basado en la teoría de conjuntos difusos (Duch et al., 2004). En la tabla 1.1 se muestra el conjunto de símbolos utilizados en el proceso de sumarización lingüística de datos, adaptada de (Piñera Trinchet, 2013).

Tabla 1.1: Simbología usada en la sumarización lingüística

Símbolos	**Significado**
D	Base de datos
Y	Conjunto de todos los objetos de la base de datos
M	Cantidad de objetos en Y
Y_m	El m-ésimo objeto en la base de datos
V_m	El nombre del n-ésimo atributo
X_n	El dominio de V_n
x V	El conjunto de todos los nombres de los atributos
V_n^m	El valor n-ésimo del atributo para Y_m
d_m	Todos los valores de los atributos para Y_m
S_n Q	Sumarizador Cuantificador
R	Cualificador
T_1	El valor de verdad
T_2	El grado de imprecisión
T_3	El grado de cobertura
T_4	El grado de adecuación
T_5	Longitud del resumen

Esta técnica introducida por Yager (Yager, 1982), puede definirse como una herramienta intuitiva para descubrir conocimiento basada en el lenguaje natural y humanamente consistente. La Sumarización Lingüística de Datos permite que una base de datos con datos numéricos en sus atributos sea resumida con respecto a uno o varios atributos por medio de proposiciones lingüísticamente cuantificadas del modo:

T (*La mayoría de los empleados jóvenes y bien calificados son bien pagados*) $= 0.7$.

Teniendo un conjunto de objetos $Y = \{y_1, y_2, y_3, \cdots, y_n\}$ que posean un atributo V, el conjunto $D =$

$\{V(y_1), \cdots, V(y_n)\}$ contendría el valor de V para los n objetos de Y. Un resumen lingüístico de los datos agrupados en D consiste en una o varias proposiciones de la forma: $QRy'areS = T$, dónde:

Un sumarizador (S) es una expresión lingüística representada por un conjunto difuso (término lingüístico de un atributo difuso). Ejemplo: *bajo* para el atributo *salario*.

Un cuantificador (Q) (absoluto o relativo), es una medida que indica el rango de datos que satisfacen a S y se asume que es un término lingüístico representado por un conjunto difuso en el intervalo $[0, 1]$ como se muestra en la ecuación 1.1. Ejemplo: *alrededor de 5 / la mayoría*.

$$\mu_Q(x) = \begin{cases} 1 & \text{para } x \geq 0{,}8 \\ 2x - 0{,}6 & \text{para } 0{,}3 < x < 0{,}8 \\ 0 & \text{para } x \leq 0{,}3 \end{cases} \tag{1.1}$$

Un Truth (T) representa el grado de validez de la proposición. Asume un valor entre [0,1]. Existen varios enfoques para determinar T, aunque dos de ellos han sobresalido. El primero es el propuesto por Zadeh (Zadeh, 1983) y que consiste en el cálculo difuso de cuantificadores lingüísticos. Luego Yager (Yager, 1988) introdujo los operadores OWA. Además de estos, otros métodos han sido planteados, entre ellos las medidas propuestas por el propio Yager en su trabajo fundacional (Yager, 1982), así como las de George y Srikanth (George y Srikanth, 1996) y Chen (Chen et al., 2001). Más adelante se analizarán con más detalles las cinco medidas propuestas por (Kacprzyk y Zadrożny, 2010; Kacprzyk y Yager, 2001; Kacprzyk et al., 2000).

Opcionalmente puede aparecer un **cualificador (R)**. Este es otra expresión lingüística que determina un subconjunto de objetos dentro de la base de casos (filtro difuso). Ejemplo: *joven* para el atributo *edad*.

Forma de calcular T

Como se mencionó anteriormente existen diversas maneras de calcular el grado de validez de una proposición. A continuación serán analizados las medidas propuestas por (Kacprzyk y Zadrożny, 2010; Kacprzyk y Yager, 2001; Kacprzyk et al., 2000; Zadeh, 1983).

Zadeh propone calcular el valor de T según las ecuaciones:

$$T(Qy'areS) = \mu_Q \left[\frac{1}{n} \sum_{i=1}^{n} \mu_S(y_i) \right] \tag{1.2}$$

$$T(QRy'areS) = \mu_Q \left[\frac{\sum_{i=1}^{n} (\mu_R(y_i) \bigwedge \mu_S(y_i))}{\frac{1}{n} \sum_{i=1}^{n} \mu_R(y_i)} \right] \tag{1.3}$$

donde n es el número de casos en la base de datos y el símbolo $\bigwedge$ denota el mínimo entre $\mu_R(y_i)$ y $\mu_S(y_i)$, (una t-norma).

Kacprzyk y sus colegas consideran que esta propuesta a pesar de ser correcta no comprende todos los aspectos de un resumen lingüístico y proponen cinco criterios para determinar el valor de certeza de la proposición. Estos criterios son: a) valor de verdad, que se corresponde con lo propuesto por Zadeh, b) grado de imprecisión, c) grado de cobertura, d) grado de adecuación y e) longitud del resumen y son calculados de la manera que sigue.

T_2 para un conjunto borroso

$$T_2 = 1 - \sqrt[m]{\prod_{j=1,2,\cdots,m} in(S_j)} \tag{1.4}$$

donde $in(S_j)$ se calcula para cada conjunto difuso S_j que esté presente en el resumen y se optiene según la ecuación 1.5.

$$in(S_j) = \frac{card\left\{x \in X_j : \mu_{S_j} > 0\right\}}{card\ X_j} \tag{1.5}$$

Dada la descripción de (sumarizador) S, como una familia de conjuntos difusos $S = s_1, s_2, ..., s_m$, o sea, las etiquetas que representan a cada conjunto difuso de ese resumen. El término *card* denota la cardinalidad del correspondiente conjunto no difuso, lo que se traduce al plano del conjunto difuso s_j donde $in(s_j)$ posee el valor más alto.

El grado de cobertura (T_3) expresa cuántos objetos y_i que cumplen con R son cubiertos por S y se define como:

$$T_3 = \frac{\sum_{i=1}^{n} t_i}{\sum_{i=1}^{n} h_i} \tag{1.6}$$

donde

$$h_i = \begin{cases} 1 & \text{si } \mu_R(y_i) > 0 \\ 0 & \textit{en otro caso} \end{cases} \tag{1.7}$$

$$t_i = \begin{cases} 1 & \text{si } \mu_S(y_i) > 0 \textit{ y } \mu_R(y_i) > 0) \\ 0 & \textit{en otro caso} \end{cases} \tag{1.8}$$

Para presentar el grado de adecuación (T_4) se supone que el resumen contiene la descripción de los conjuntos difusos que están presentes en él. Este es particionado en m resúmenes parciales, cada uno de los cuales abarca los atributos particulares $S_1, S_2, ..., S_m$, de manera que cada resumen parcial corresponde a un valor difuso solamente (S_j), donde $S_j(y_i) = \mu_{S_j}(y_i)$. El valor T_4 es calculado como:

$$T_4 = abs\left(\prod_{j=1,\cdots,m} r_j - T_3\right) \tag{1.9}$$

donde

$$r_j = \frac{\sum_{i=1}^{n} h_i}{n},\ j = 1, \cdots, n \tag{1.10}$$

$$h_i = \begin{cases} 1 & \text{si } S_j(y_i) > 0 \\ 0 & \textit{en otro caso} \end{cases} \tag{1.11}$$

T_4 expresa cuán característico resulta el resumen para la base de casos en particular.

Finalmente, la longitud del resumen (T_5) es obtenida a través de la ecuación 1.12 y su relevancia se le atribuye a que un resumen largo no es fácilmente comprensible por el ser humano.

$$T_5 = 2(0{,}5^{cardS}) \tag{1.12}$$

Donde card S, es el número de elementos en S.

Luego del cálculo de estos criterios proponen determinar el grado de validez total del resumen (T) como sigue:

$$T = T(T_1, T_2, T_3, T_4, T_5; w_1, w_2, w_3, w_4, w_5) = \sum_{j=1,\cdots,5} w_j \cdot T_i \tag{1.13}$$

donde $w_1, w_2, w_3, w_4, w_5 \in [0,1]$ cumpliendo con que

$$\sum_{j=1,\cdots,5} w_j = 1 \tag{1.14}$$

y el problema consiste en encontrar un resumen óptimo $S^* \in S$, es decir, el mayor valor de

$$\sum_{j=1,\cdots,5} w_j \cdot T_i \tag{1.15}$$

Los pesos w_1, w_2, w_3, w_4, w_5 se pueden predefinir por parte del usuario.

1.1.7. Protoformas

Las protoformas (Zadeh, 2002) son prototipos abstractos de los resúmenes que se desea obtener, y pueden tener la siguiente composición: $QRy's are S$. Estas protoformas pueden formar jerarquías, situándose en el nivel más alto a las más abstractas. Las más abstractas son aquellas en las que no se hacen suposiciones importantes en torno al resumen buscado (Kacprzyk y Zadrożny, 2010, 2005).

Los casos límites *mayor y menor abstracción* son aquellos en los que se asume que la protoforma es totalmente abstracta o se supone que todos los elementos de la protoforma son términos lingüísticos específicos previamente conocidos. La tabla 1.2 muestra los seis tipos básicos de protoformas, ordenados de menor a mayor nivel de abstracción. Véase que en el tipo 0 se considera aquella en la que se conocen todos los elementos de la protoforma y solo se calcula el valor de T.

Tabla 1.2: Clasificación de los resúmenes lingüísticos (adaptado de (Kacprzyk y Zadrożny, 2010)).

Tipo	**Protoforma**	**Dado**	**Busca**
0	QRy's are S	Todo	$Validez(T)$
1	Qy's are S	S	Q

Continuación en la próxima página

Tabla 1.2 – *Continuación de la página anterior*

Tipo	**Protoforma**	**Dado**	**Busca**
2	QRy's are S	S y R	Q
3	Qy's are S	Q y $S_{structura}$	S_{valor}
4	QRy's are S	Q, R y $S_{structura}$	S_{valor}
5	QRy's are S	Nada	S, R y Q

1.2. Ejemplo de LDS utilizando lógica difusa

Con el propósito de comprender la aplicación práctica de los conceptos analizados en las sección (1.1.3 y 1.1.4), seguidamente se presenta un ejemplo de LDS utilizando lógica difusa, adaptado a partir de lo presentado por los principales autores de este tema en (Kacprzyk y Yager, 2001; Kacprzyk y Zadrożny, 2005).

Se tiene una base de datos de las ventas de una tienda de informática, cuya estructura básica es la Tabla (1.3). En la Tabla (1.4) pueden observarse las relaciones (*resúmenes*) entre varios atributos, unido a sus valores de T_1, T_2, T_3, T_4 y su grado total de validez T. Los valores de T_5 *longitud* no son analizados. Este es un ejemplo de la forma más sofisticada de resúmenes lingüísticos que pueden obtenerse, dado que además del resumidor S y el cuantificador Q en varios casos aparecen dos cualificadores R por ejemplo *fecha y tiempo*. No se presenta sólo un resumen lingüístico, sino un conjunto de ellos que son los de mayor grado de validez y dan mucha información al usuario *el analista* sobre las relaciones entre los atributos seleccionados; por otra parte, son simples y humanamente consistentes. Además, el uso de protoformas proporciona una universalidad muy necesaria y puede simplificar en gran medida el diseño conceptual y algorítmico, y por tanto su aplicación.

Tabla 1.3: Estructura básica de la base de datos, tomado de (Kacprzyk y Zadrożny, 2010), traducción propia.

Nombre atributo	**Tipo atributo**	**Descripción**
Fecha	Date	Fecha de la venta
Tiempo	Time	Tiempo de operación de venta
Nombre	Text	Nombre del producto
Cantidad (número)	Numeric	Número de productos vendidos en la transacción
Precio	Numeric	Precio unitario
Comisión	Numeric	Comisión en venta (en por ciento)
Valor	Numeric	Valor = cantidad (número) × precio del producto

Continuación en la próxima página

Tabla 1.3 – *Continuación de la página anterior*

Nombre atributo	**Tipo atributo**	**Descripción**
Descuento	Numeric	Descuento para la transacción (en por ciento)
Grupo	Text	Grupo de productos al que pertenece el producto
Valor de la transacción	Numeric	Valor de la transacción completa
Total de venta al cliente	Numeric	Valor total de las ventas al cliente en el año fiscal
Frecuencia de compras	Numeric	Número de compras por cliente en el año fiscal
Ciudad	Text	Ciudad donde vive el cliente

Tabla 1.4: Resúmenes lingüísticos que expresan las relaciones entre los atributos: tamaño del cliente, la regularidad del cliente, fecha de venta, el tiempo de la venta, comisiones, grupos de productos y los días de venta. Adaptado de (Kacprzyk y Zadrożny, 2010, 2005), traducción propia.

Resumen	Grado de verdad (T_1)	Grado de imprecisión (T_2)	Grado de cobertura (T_3)	Grado de conformidad (T_4)	Promedio ponderado (validez total T)
Muchas ventas el sábado son alrededor del mediodía con una comisión baja	0.3951	0.2748	0.6591	0.3843	0.3863
Muchas ventas el sábado son alrededor del mediodía para los clientes más grandes	0.4430	0.4075	0.7500	0.3425	0.3648
Muchas ventas el sábado son alrededor del mediodía	0.4654	0.4708	0.7841	0.3133	0.3564

Continuación en la próxima página

Tabla 1.4 – *Continuación de la página anterior*

Resumen	Grado de verdad (T_1)	Grado de imprecisión (T_2)	Grado de cobertura (T_3)	Grado de conformidad (T_4)	Promedio ponderado (validez total T)
Muchas ventas el sábado son alrededor del mediodía para los clientes habituales	0.4153	0.3540	0.6932	0.3391	0.3558
Algunas ventas para clientes habituales son con comisión baja	0.1578	0.5837	0.1954	0.3882	0.3451
Algunas ventas para pequeños clientes son con comisión baja	0.1915	0.5837	0.2263	0.3574	0.3263
Algunas ventas para los clientes de una sola vez son con comisión baja	0.1726	0.5837	0.2339	0.3497	0.3195
Muchas ventas para pequeños clientes son para clientes no habituales	0.5105	0.1458	0.7709	0.6250	0.5986

1.3. Estado actual de la Informática Jurídica en el mundo y en Cuba

En esta sección se hace un acercamiento al desarrollo actual de la Informática Jurídica. Se abordan los aspectos relacionados con las clasificaciones de esta y de los sistemas informáticos jurídicos que se mencionan en el epígrafe 1.1.2. Además se hace un estudio enmarcado en el origen y evolución de la informática jurídica así como las tendencias y el desarrollo de los sistemas jurídicos en el mundo y en Cuba.

Estado actual de la Informática Jurídica en el mundo

La Informática, como uno de los fenómenos más significativos de los últimos tiempos, deja sentir su incontenible influjo en prácticamente todas las áreas del conocimiento humano, dentro de las cuales el

Derecho no es la excepción, lo que da lugar, en términos instrumentales a la llamada Informática Jurídica. Esta área del conocimiento surgió en 1959 en Estados Unidos y ha sufrido cambios afines a la evolución general de la misma Informática (LLamas, 2008).

En sus primeros años, la Informática Jurídica se presentó en los términos de una Informática Documentaria de carácter jurídico, es decir, creación y recuperación de información que contenía datos jurídicos (leyes, jurisprudencia, doctrina) o al menos de interés jurídico. Poco a poco se empezó a vislumbrar la idea de que en estos bancos de datos jurídicos se podían obtener no sólo información sino también, mediante programas estudiados expresamente, verdaderos actos jurídicos, como certificaciones, atribuciones de juez competente y sentencias pre modeladas, por lo que nació así a fines de los años sesenta, la llamada Informática Jurídica de Gestión (Téllez, 1998).

Como la información y procedimientos eran fidedignos y permitían llegar a buenos resultados, surgió lo que hoy es considerado por algunos tratadistas como los sistemas expertos legales (Informática Jurídica Metadocumentaria) (Téllez, 1998).

A principios de 1966, doce estados de la Unión Americana se propusieron desarrollar un sistema interno de recuperación de documentos legales. Para el año 1969 fue desarrollado por la Universidad de Pittsburgh el sistema LITE hoy llamado FLITE, bajo contrato con la Fuerza Aérea Norteamericana. Para 1967 comenzó a desarrollarse el sistema OBAR de Data Corporation y para 1973 el sistema LEXIS, sucesor del OBAR luego de la fusión de Data Corporation con Mead Corporation, ambos sistemas basados en términos de informática documentaria y de gestión. Actualmente el sistema LEXIS no solo opera en Estados Unidos, también en países de europa como Inglaterra y Francia donde se han almacenado las resoluciones del Consejo de Estado (Olivera, 2010).

Con el surgimiento y desarrollo de la Inteligencia Artificial (I.A), se han venido creando disímiles sistemas computarizados con la finalidad de reducir el esfuerzo del hombre. Varios ejemplos se encuentran en el área de control de sistemas, planificación automática, etc. Por citar un ejemplo, cabe mencionar, en la robótica, los teleoperadores, robots de servicio e industriales. Los sistemas de I.A actualmente son parte de la rutina en campos como economía, medicina e ingeniería.

Los sistemas computacionales, en la inteligencia artificial, deben ser capaces de simular características que son comúnmente asociadas con la inteligencia de la conducta humana. Un sistema inteligente es aquel que exhibe un comportamiento similar al humano cuando se enfrenta a un problema idéntico y no seamos capaces de distinguir entre un ser humano y un programa de computadora en una conversación a ciegas. En el área del Derecho, la I.A ha encontrado su lugar en el área de la Informática Jurídica Metadocumentaria. Esta se traduce en los sistemas expertos legales constituidos por una base de conocimientos, mecanismos de inferencias y la interfaz entre el usuario y la máquina (Martínez Bahena, 2012).

En la práctica del Derecho, la búsqueda del conocimiento jurídico está orientada a resolver cuestiones con consecuencias en la vida política. La Informática Jurídica ha comenzado a ocuparse también del campo de la decisión que es, sin lugar a dudas, el que más dificultad presenta. No es necesario que el sistema tome la decisión; simplemente puede ayudar a la decisión que se puede dar en varios planos y niveles (Téllez, 1998).

Con base en las definiciones generalmente aceptadas, se considera que un sistema experto es aquel sis-

tema computacional capaz de proporcionar respuestas que, atribuidas a los humanos, presuponen procesos inteligentes de carácter heurístico (no algorítmicos). Los sistemas expertos tratan de emular la toma de decisiones de un experto humano en dominios de conocimiento bien definidos. El término *pensar* denota una serie de fenómenos neurofisiológicos que no podemos encontrar en las máquinas, pero si se puede hacer que una máquina simule algunos procesos de pensamiento (Martínez Bahena, 2012).

Los tipos de Sistemas Expertos Jurídicos se distinguen como sigue de acuerdo a (Luño, 1996):

1. Sistemas Expertos Jurídicos para la recuperación inteligente de documentación jurídica. Los sistemas de informática jurídica documental han ido evolucionando de tal forma que han alcanzado un grado de dificultad que hacen que el usuario necesite unos conocimientos técnicos para su utilización que realmente complican su manejo. Para eso se crean los Sistemas Expertos para la recuperación inteligente de documentación jurídica, como, por ejemplo, el sistema EASYFIND (Palmirani y Brighi, 2003).

2. Los Sistemas Expertos Jurídicos Hypertextuales. Se trata de programas de software capaces de gestionar el texto completo o parcial de documentos, así como la red de relaciones y proyecciones de los mismos. Estos Sistemas Expertos tienen como función estructurar información, de tal forma que ante una petición de un usuario sobre una determinada información, se suministre ésta y además toda la relacionada con la misma, de tal forma que se tenga acceso a lo solicitado y a lo relacionado, pero adicionalmente, según el usuario vaya solicitando informaciones sucesivas, el hypertexto va a relacionar todas las consultas y suministrar toda la información común a todas ellas si se le solicita. Como ejemplo tenemos el sistema ELP-ADVISOR (Fameli et al., 1991).

3. Los Sistemas Expertos para el dictamen jurídico. Es el sistema experto por excelencia que califica jurídicamente un determinado supuesto o ayuda a interpretar las normas jurídicas aplicables al mismo. Dentro de estos sistemas expertos cabe destacar el sistema TAXMAN (McCarty, 1977) en sus variantes I y II dedicada a regímenes fiscales, el Sistema MIT (Meldman, 1975), cuya función es dar informaciones sobre decisiones judiciales en asuntos de agresiones y violencias.

4. Los Sistemas Expertos Legislativos. Las funciones de estos sistemas expertos pueden influir en tres aspectos de la legislación:

   - En la técnica de redacción de textos normativos *legal drafting*. Consiste en que suministrando al sistema experto un texto en lenguaje normal lo convierte en un texto normativo.
   - El control del proceso legislativo *legal process*, consistente en que el sistema experto va a ser capaz de detectar las antinomias, reiteraciones y lagunas en la ley redactada.
   - En la planificación del sistema legislativo (legal system). Por medio de estos sistemas expertos se permite evaluar el impacto de nuevas normas en el sistema jurídico y/o en el sistema social.

5. Sistemas Expertos destinados a la enseñanza del Derecho. Las funciones que tendría este tipo de sistema experto es la de orientar al estudiante de Derecho en su aprendizaje, así como ayudar al

> profesor en las calificaciones, de tal forma que el docente va a programar el sistema experto al nivel que se va a exigir al alumno, y el profesor lo que realmente va a hacer es ayudar para que se puedan superar esos límites, pudiendo dedicar del enseñante más tiempo a la labor docente e investigadora.

El atractivo de un sistema experto es fundamentalmente su disponibilidad y conveniencia. A diferencia de un humano que tiene que dormir, comer, descansar, tomar vacaciones, etcétera, el sistema experto está disponible durante las veinticuatro horas del día durante todos los días del año. Además, pueden crearse muchos sistemas expertos, mientras que hay un número limitado de expertos humanos. A diferencia de los humanos, el experto computarizado nunca muere llevándose sus conocimientos con él. Los conocimientos de un sistema experto pueden ser copiados y almacenados fácilmente, siendo muy difícil la pérdida de éstos (Martínez Bahena, 2012).

Estado actual de la Informática Jurídica en Cuba

Esta subsección está basada en un artículo realizado por Yarina Amoroso (Amoroso Fernández, 2015). Se contempla principalmente, el trabajo de esta autora debido a que constituye una autoridad en este tema en el ámbito nacional y regional.

La expresión *Sector Jurídico Cubano* en su sentido amplio, comprende a los órganos y entidades cuya labor fundamental se corresponde con una función jurídica como el Ministerio de Justicia, el sistema de Tribunales, la Fiscalía General de la República de Cuba y la Organización Nacional de Bufetes Colectivos. En un sentido extendido, el concepto permite enmarcar un área *muy extensa* del trabajo jurídico en organismos como el Ministerio de Trabajo y Seguridad Social, que como el resto de los que integran la Administración Central del Estado, cuentan con organizaciones específicas de asesoría jurídica que por lo general centran el trabajo especializado en materia de Derecho acorde con las funciones de su competencia.

Para entender el esfuerzo de informatización del sector jurídico cubano, se hace necesario enmarcarlo en el tiempo para identificar los momentos cruciales de su desarrollo. Este esfuerzo tiene sus inicios en el I Taller de Informática Jurídica en 1991. Aquí se puso en evidencia que para impulsar la informatización en el Sector Jurídico se dependía del establecimiento de una política en el que se definieran los objetivos y metas a alcanzar por cada uno de los órganos y organismos jurídicos. Se identificó también la necesidad de garantizar la preparación del personal y en ese sentido se organizaron eventos territoriales, se incrementó la preparación que en informática recibieron los graduados universitarios de Derecho y se realizaron eventos nacionales e internacionales con la participación de numerosos juristas cubanos.

Fruto de este esfuerzo nació en 1995 la Sociedad Cubana de Derecho e Informática, espacio académico que mantiene su rol de convocatoria para la reflexión, capacitación y articulación de proyectos desde la Unión Nacional de Juristas en materia de informatización del sector jurídico, en armonía con los requerimientos concretos de cada institución.

En el V Congreso de la Unión Nacional de Juristas de Cuba se señaló:

> "si se relaciona con el nivel alcanzado y ritmo de los trabajos de informatización de la sociedad cubana, se puede llegar a la conclusión de que el Sector Jurídico, no obstante, el esfuerzo de organismos e

> instituciones, que es innegable se viene haciendo, se encuentra rezagado, por lo que es de importancia superar esta situación".

Para esto, se acuerda en el congreso trabajar porque cada institución del Sector tenga una plantilla informática adecuada y lograr su completamiento. Igualmente gestionar y obtener de los ministerios de Educación Superior y Economía y Planificación la asignación de graduados de las especialidades de informática. También fortalecer los vínculos de la actividad informática que se desarrolla en el sector con la labor formadora de las universidades, priorizando aquellos órganos e instituciones de la práctica jurídica que posean categoría de Unidades Docentes, así como fortalecer el papel que los altos centros de estudio deben jugar en la transmisión de los conocimientos informáticos y computacionales.

El en año 1997 a propósito del V Congreso del Partido Comunista de Cuba se aprueban, por primera vez, los Lineamientos Generales para la Informatización de la Sociedad y la forma de implementarlos con objetivos generales hasta el 2000.

A esta primera versión le sucedieron revisiones sistemáticas que precisan siete áreas de acción, a saber:

- Infraestructura, Tecnologías y Herramientas
- Formación Digital
- Fomento de la Industria Nacional de las Tecnologías de la Información y las Comunicaciones
- Investigación, Desarrollo y Asimilación de Tecnologías
- Utilización de las TIC en la Dirección
- Sistemas y Servicios Integrales para los ciudadanos
- Utilización de las TIC en el Gobierno, la Administración y la Economía

Ello catalizó los esfuerzos del Sector Jurídico cubano en materia de informatización y se consolidaron especialmente las estructuras administrativas de informática en el seno de las instituciones jurídicas participando en las reuniones metodológicas al tiempo que se facilitó la inserción del sector en programas priorizados. Desde entonces hasta la fecha el sector jurídico ha estado presente como parte del esfuerzo de informatización de la Sociedad Cubana.

En el año 2004 a partir de una reunión conjunta del Consejo de Dirección del Ministerio de Justicia de Cuba y la Rectoría de la Universidad de Ciencias Informáticas se cimienta una comprometida relación de la UCI con el sector. Así se inició una relación en doble dirección, que sin desconocer valiosas y avanzadas experiencias internacionales *que sirven de referencia para ser estudiadas*, se asumió que la informatización del Sector Jurídico debía desarrollarse a partir de nuestras propias experiencias y tomar en cuenta los objetivos respondiendo al contenido y esencia de nuestro Derecho.

Es menester señalar que si se retoma el criterio amplio de sector jurídico en correspondencia con funciones que implican relaciones jurídicas se puede expresar también los sistemas informáticos desarrollados para la Cámara de Comercio de Cuba y el Consejo de Patrimonio Cultural en el año 2013, por solo citar dos ejemplos de resultados del Centro de Gobierno Electrónico de la Facultad 3.

A este esfuerzo se le sumaron otros empeños con parte del núcleo central del sector jurídico cubano como los son los proyectos de Fiscalía General de la República y del Sistema de Tribunales de Cuba desplegados en el año 2014 y 2015 respectivamente. Esfuerzo que aún continúa y amplia.

Como parte del proceso de informatización del Sistema de Tribunales cubanos y de la Fiscalía General de la República de Cuba desarrollado de conjunto con estas instituciones y la Universidad de las Ciencias Informáticas, se han analizado los procedimientos que ejecutan estas entidades judiciales, resultando una serie de propuestas tecnológicas por parte del equipo de desarrollo para viabilizar, con el uso de la tecnología, la actividad jurídica y en especial el procedimiento jurídico.

Para ello, se ha compartido el objetivo de desarrollar un sistema de gestión que garantice la estandarización y celeridad en la realización de los procedimientos, contribuyendo a la toma de decisiones de organismos superiores u otros entes dependientes e involucrados.

Como es apreciable, la informatización en el sector jurídico cubano se encuentra rezagado con respecto al desarrollo de las T.I.C en el mundo y específicamente dentro del área de Informática jurídica mundial. También es apreciable que hace ya más de una década que se viene trabajando sobre la base del desarrollo de la informatización del sector jurídico cubano y con esfuerzos propios.

Cabe mencionar aquí la Tesis de Diploma (Hidalgo García y Otero Morfa, 2015). En este trabajo se expone la creación de los mercados de datos para los departamentos de Procesos Penales y Gestión de Cuadros y Personal de Apoyo de la Fiscalía General de la Repúblicade Cuba (FGR). El objetivo de estos mercados de datos es facilitarle a la FGR una forma más eficiente de poder almacenar la información que se gestiona en los departamentos antes mencionados para su explotación y análisis.

Otra evidencia del desarrollo de la Informática Jurídica en nuestro país lo contituye (Marciel, 2015). En este trabajo, se brinda una solución que permite obtener reglas de asociación que definan relaciones entre los atributos registrados de los expedientes de los procesos ordinarios contractuales de la Sala de lo Económico del Tribunal Provincial de La Habana. Se facilita así, la toma de decisiones a nivel de país.

1.4. Estado actual del desarrollo de la LDS en el Mundo y en Cuba

Los nuevos modelos de procesamiento y control basados en lógica borrosa, junto con algunos otros de relativa novedad, se engloban dentro de las denominadas técnicas de *soft computing*. Estas nuevas técnicas se inspiran en las soluciones que la naturaleza ha encontrado durante millones de años de evolución, a numerosos problemas tecnológicos que involucran el tratamiento de cantidades masivas de información, redundante, imprecisa y ruidosa (Matías y Vicente, 2008).

Desarrollo actual de la LDS en el mundo

Los sistemas basados en lógica borrosa se vienen utilizando en aplicaciones de diversas índoles (Matías y Vicente, 2008). La sumarización lingüística, como técnica de minería de datos, constituye una herramienta capaz de proporcionar resúmenes destinados a capturar características esenciales de los datos originales de acuerdo con las necesidades del usuario. Por tanto, estos resúmenes, proporcionan una visión más simple de los datos recogidos en una base de datos.

Así, (Dıaz-Hermida y Bugarın, 2010) describe como se puede generar un conjunto de expresiones dotadas de una semántica que es conveniente y completa para el consumo humano para resumir datos lingüísticamente. Expone como un modelo con cuantificadores difusos probabilísticos puede ser usado para construir resúmenes difusos cuantificados.

Otro enfoque para la obtención de resúmenes lingüísticos de datos son los resúmenes con estados cuantificados, a los cuales se les asocia un grado de validez que tiene un significado claro en términos de cantidad y la calidad y que proporciona una descripción precisa de la información almacenada en la base de datos (Liétard, 2012). Todo esto proporciona una herramienta de apoyo a la toma de decisiones.

Siguiendo la misma idea, se propone un enfoque para la obtención de resúmenes lingüísticos a partir de las tendencias que pueden estar presentes en los datos temporales, siendo el análisis de tendencias de gran importancia y ampliamente utilizado en muchos campos, como la climatología y la economía (Kacprzyk et al., 2006).

Debido al papel principal que el tiempo juega en general, la mayor parte de *analizar los datos* está relacionado con la dimensión de tiempo. Ejemplos bien conocidos de series temporales incluyen las tendencias de cambio de valores, la evolución de las ventas de un producto determinado a lo largo del tiempo, la afluencia de pacientes a un centro médico, la variación de los precios de un determinado producto durante un año (como ejemplo, el petróleo crudo, gasolina o los tomates), etc (Batyrshin y Sheremetov, 2008). Un acercamiento a la minería de datos de series de tiempo lo hace (Castillo-Ortega et al., 2011), en el que se propone el uso de un enfoque evolutivo para obtener resúmenes lingüísticos a partir de datos de series de tiempo. Dado el número de posibles resúmenes finales y las diferentes formas de medir su calidad ha llevado a adoptar el uso de un algoritmo evolutivo multiobjetivo.

Otro análisis de la SLD lo hace (Donis-Diaz et al., 2015) definiendo el papel que presenta un algoritmo genético mejorado, diseñado específicamente para la producción de resúmenes de datos lingüísticos. El modelo no es capaz de obtener un conjunto de *buenos resúmenes lingüísticos* pero si un *buen conjunto* de los resúmenes con una alta diversidad y buenos valores para las medidas de calidad en los resúmenes individuales. Estos resultados podrían ser interés a los expertos para analizar de forma más sencilla el comportamiento de la información en una base de datos.

La idea de Zadeh de computación con palabras y percepciones basada en su concepto de procesamiento del lenguaje natural (PLN), ha conducido a una nueva dirección en el uso del lenguaje natural en el resumen lingüístico de bases de datos. (Kacprzyk y Zadrożny, 2014) presenta brevemente una implementación para una base de datos de ventas de tiendas de informática como un convincente ejemplo de que estas herramientas y técnicas son aplicables y funcionales. Estos resúmenes incluyen datos de una base de datos interna de la empresa y datos descargados desde bases de datos externas a través de Internet.

Desarrollo actual de la SLD en Cuba

En Cuba también se ha hecho un acercamiento a la sumarización lingüística de datos, principalmente en la Universidad Central de Las Villas. Algunos de los trabajos más recientes están enmarcados en el análisis de datos de fluencia para el diseño de nuevos aceros. Seguidamente se comentan brevemente algunos de

ellos.

Conjugando con el uso de los algoritmos genéticos (Donis-Díaz et al., 2014) propone un modelo híbrido del algoritmo genético con la búsqueda local para descubrir resúmenes lingüísticos y su aplicación en el análisis de los datos de fluencia, que proporciona resúmenes de alta calidad y una amplia gama de información y supone una mejora en los resultados en comparación con los obtenidos utilizando el modelo clásico de algoritmo genético. El éxito de los resúmenes lingüísticos para la descripción de las tendencias de los datos de fluencia ha sido expuesto en (Díaz et al., 2011).

Algunos modelos utilizando metaheurísticas basadas en un procedimiento de *mejora de las soluciones*, específicamente algoritmos genéticos (AG), se han propuesto con anterioridad para la sumarización lingüística de datos numéricos. Los autores de (Donis-Díaz et al., 2015) proponen en su trabajo un nuevo modelo de SLD basada en la optimización de Colonia de Hormigas (ACO) y una metaheurística que utiliza un procedimiento de *construcción de la solución*. Ambos modelos se comparan en SLD sobre los datos de fluencia. Los resultados muestran cómo el modelo basado en ACO supera las medidas de bondad del resumen final, pero no mejora los resultados del modelo basado en AG en relación con la diversidad del resumen.

1.5. Estado actual de aplicación de la SLD a la Informática Jurídica en el mundo y en Cuba

Como se puede apreciar en las secciones 1.1.2 y 1.3 el desarrollo de la informática, y con esta la Informática Jurídica, ha permitido la creación de software que constituyen una herramienta de apoyo para las tareas que realizan hoy en día los juristas en todo el mundo. Estos sistemas, poseen grandes bases de datos en las que se almacena información de datos jurídicos como leyes, jurisprudencia, doctrina, así como información de hechos y actos procesales, etc.

La SLD, como técnica de minería de datos, contituye una herramienta factible para extraer patrones a partir de la información almacenada en bases de datos. La sumarización lingüística permite captar y describir brevemente las tendencias y características que aparecen en un conjunto de datos numéricos o no, a través de resúmenes generalmente cortos, como una frase. La información que proporcionan estos resúmenes puede ser utilizada como apoyo a la toma de decisiones.

Recientemente (Marciel, 2015), utiliza técnicas de Minería de Datos e IA para revelar patrones no triviales y facilitar la toma de decisiones a nivel de país. Los datos que se utilizan en el marco de este trabajo, provienen de diversas fuentes, con formatos nominales y numéricos que se fusionan con el objetivo de resumir la información. Para esto, se aplican los algoritmos Apriori y Predective Apriori para obtener reglas de asociación que definan relaciones entre los atributos registrados de los expedientes de los procesos ordinarios contractuales de la Sala de lo Económico del Tribunal Provincial de La Habana en los años 2012, 2013, 2014 y primer trimestre del 2015.

1.6. Metodología de desarrollo de software

Una metodología de desarrollo de software es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos (Sommerville y Galipienso, 2005).

En la construcción del software actual existen dos corrientes conocidas como enfoque ágil y enfoque prescriptivo de desarrollo. El enfoque prescriptivo, denominado en algunas bibliografías como tradicional o pesado, busca la estructura, orden y consistencia del proyecto de desarrollo de software en cuestión. Se les llama prescriptivos porque prescriben un conjunto de elementos del proceso (acciones, tareas, productos de trabajo, mecanismos de control y aseguramiento de la calidad). Además definen la forma en que los elementos del proceso mencionados anteriormente deben relacionarse entre sí (Sommerville y Galipienso, 2005).

El enfoque ágil, llamado también como enfoque ligero, se centra en los miembros del equipo y su interacción, en la entrega rápida de versiones de software funcional, en la colaboración constante del cliente y la facilidad para manejar los cambios, dándole menor importancia a las herramientas, documentación, la formalidad y planificación exhaustiva del proceso (Beck et al., 2001). Aunque estas visiones parezcan opuestas, lo cierto es que se requiere disciplina, pero también adaptabilidad y agilidad. La selección de un enfoque y en función de este la metodología a utilizar, dependen de las circunstancias y características específicas de cada proyecto de desarrollo de software (Velázquez, 2012).

La tabla 1.5 muestra la de comparación entre las metodologías ágiles: Programación Extrema (XP) (Beck, 2000), SCRUM (Schwaber, 2004), Método de Desarrollo de Sistemas Dinámicos (DSDM) (Stapleton, 1997) y Desarrollo Adaptativo de Software (ASD) (Ghani, 2000) de acuerdo a su uso, equipo de desarrollo, enfoque de desarrollo, roles y fases de cada una.

Tabla 1.5: Comparación entre metodologías ágiles. Tomado de (Pressman, 2010).

Metodologías				
Criterios	XP	SCRUM	DSDM	ASD
Uso	Guía equipos de desarrollo en ambientes imprecisos y cambiantes.	Propicia la colaboración eficaz. Emplea un conjunto de reglas y artefactos definiendo roles.	Es un marco de trabajo diseñado para entregar la solución correcta en el momento correcto.	Tiene como fundamento la teoría de sistemas adaptativos complejos.
Equipo de desarrollo	Equipos pequeños entre 2 y 20 miembros	Cualquier tamaño.	Cualquier tamaño.	Equipos pequeños entre 5 y 9 miembros.
Enfoque de desarrollo	Iterativo incremental	Iterativo incremental.	Iterativo	Iterativo

Continuación en la próxima página

Tabla 1.5 – *Continuación de la página anterior*

Criterios	**XP**	**SCRUM**	**DSDM**	**ASD**
Roles	Programador, Cliente, Equipo de desarrollo y Probador.	Scrum master, Dueño del producto, Equipo de desarrollo.	Director del proyecto, Arquitecto, Equipo de desarrollo y Probador.	Cliente, Líder del proyecto, Programador, Desarrollador y Probador.
Fases	Análisis, Diseño, Implementación y Pruebas.	Planeación, Puesta en escena, Desarrollo y Entrega.	Estudio de viabilidad, Estudio del negocio, Modelado funcional, Diseño, Construcción e Implementación.	Especulación, Colaboración y Aprendizaje.

Las metodologías ágiles dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos cambiantes y cuando es necesario reducir drásticamente los tiempos de ejecución pero manteniendo una alta calidad. Por tal motivo se selecciona este enfoque y se persigue una metodología ágil para el desarrollo de la herramienta.

Una vez realizada la investigación acerca de las metodologías ágiles (XP (Beck, 2000), SCRUM (Schwaber, 2004), DSDM (Stapleton, 1997), ASD (Ghani, 2000)), se opta por XP para guiar el proceso de desarrollo de software, dada las características del proyecto, la metodología posee las características suficientes y necesarias para esto. XP es una metodología que consiste en una programación rápida que se identifica por tener como parte del equipo de desarrollo al usuario final y se cuenta con un equipo de trabajo pequeño conformado por solo un integrante. Esta presenta un modelo en el que se describen los pasos a realizar en cada una de las fases que propone, facilitando así la ejecución de la aplicación. Además responde de manera eficiente a los cambios que se puedan presentar durante todo el desarrollo del software, proponiendo un ciclo de vida dinámico, y el proceso de prueba de la misma, posibilita probar cada funcionalidad al finalizar cada iteración comprobando si cumple con los requisitos iniciales.

1.7. Herramientas de software para aplicar técnicas de soft computing

Soft computing es una rama de la Inteligencia Artificial que engloba diversas técnicas empleadas para solucionar problemas que manejan información incompleta, con incertidumbre y/o inexacta. Con el creciente uso de estas técnicas se han desarrollado numerosas herramientas de software para aplicarlas. Estas herramientas son conocidas como herramientas de Minería de datos (MD). La MD consiste en el descubrimiento de conocimiento mediante la aplicación de procedimientos automáticos o semi-automáticos. Para ello se utilizan métodos de IA, aprendizaje automático, estadística y soft computing (Mathur y Nand, 2014). En la figura (1.2) se muestran las 10 herramientas más utilizadas para aplicar MD en el año 2015, según una encuesta de KDNUGGETS (KDNUGGETS, 2016) institución líder en el análisis de negocio,

procesamiento de datos y MD.

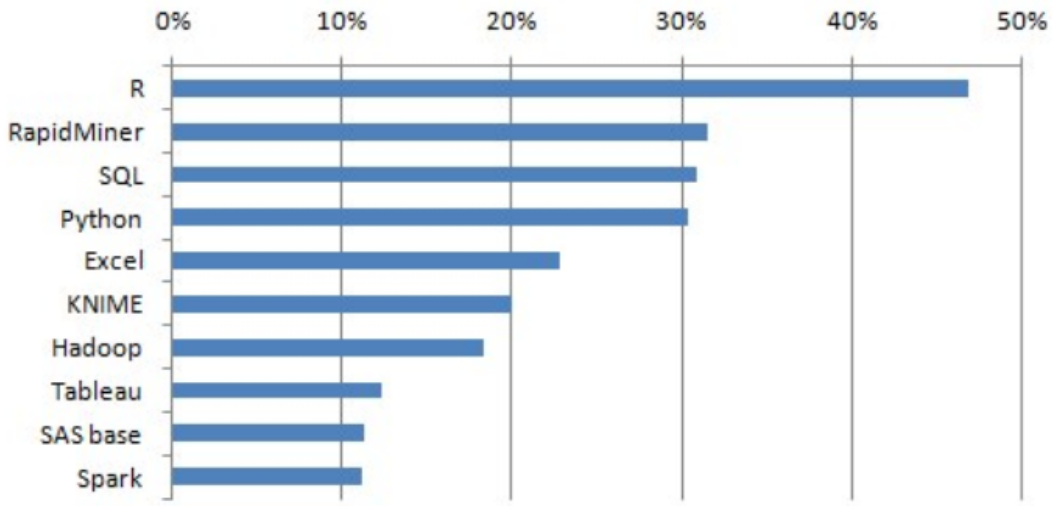


Figura 1.2: *Las 10 herramientas más utilizadas para aplicar MD.*

Como es posible apreciar R se encuentra encabezando la lista de la encuesta de KDNUGGETS, herramienta que ocupa el puesto de Rapidminer en el año 2014. Entre las características de R se puede mencionar que es un lenguaje de código abierto y un entorno para computación y gráficos estadísticos. R ofrece una amplia variedad de estadística (modelado lineal y no lineal, pruebas estadísticas clásicas, análisis de series temporales, clasificación, agrupación, ...) y es altamente extensible (Foundation, 2016). Además, R posee una alta capacidad de integración directa a nivel de funciones con bases de datos a través de lenguajes procedurales como PL/R (Conway, 2015). Por estas razones se elige R como herramienta de MD para el desarrollo de la solución propuesta.

1.8. El descubrimiento de conocimiento en bases de datos (KDD)

En el año 1996, Fayyad define KDD (del inglés knowledge discovery in database) como *el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia entendibles en los datos* (Fayyad et al., 1996). El término proceso se refiere a la secuencia iterativa de etapas o fases que lo componen. Los patrones deberían ser válidos para nuevos datos, novedosos en el sentido que deberían aportar nuevo conocimiento al dominio de aplicación y potencialmente útiles para el usuario final o tomador de decisiones.

KDD es un proceso iterativo e interactivo. Iterativo ya que la salida de alguna de las fases puede retroceder a pasos anteriores y porque a menudo son necesarias varias iteraciones para extraer conocimiento de alta calidad. Es interactivo porque el usuario, o más generalmente un experto en el dominio del problema, debe ayudar a la preparación de los datos y validación del conocimiento extraído. Aunque en el modelo de proceso KDD, Fayyad puntualiza nueve etapas para llevarlo a cabo, este se resume en las siguientes cinco fases:

- Selección de los datos sobre los que se trabajará.
- Pre-procesamiento de los datos, donde se realiza un tratamiento de los datos incorrectos y ausentes.
- Transformación de los datos y reducción de la dimensionalidad.
- Minería de datos, donde se obtienen los patrones de interés según la tarea de minería que llevemos a cabo (descriptiva o predictiva).
- Interpretación y evaluación del nuevo conocimiento en el dominio de aplicación.

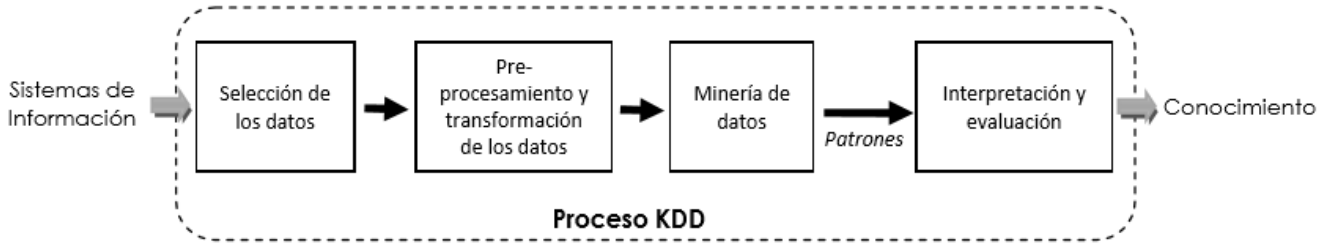


Figura 1.3: *Fases resumidas del proceso KDD, tomado de (Moine, 2013).*

Es importante destacar que, si bien KDD define las fases generales del proceso de minería de datos, no especifica qué actividades puntuales hay que realizar en cada una, quedando la definición de las mismas a criterio del equipo de trabajo.

1.9. Tecnologías para el desarrollo

En esta sección se definen las herramientas que se utilizarán para la construcción de la solución propuesta y se describen algunas de sus características. Además, se justifica su selección.

1.9.1. Sistema Gestor de Bases de Datos

PostgreSQL es un sistema de gestión de bases de datos (SGBD) objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones ofrece tantos beneficios como cualquier otra base de datos comercial. Es un sistema cien por ciento ACID, disponible para Linux, UNIX y Windows. Además posee APIs para programar en múltiples lengüajes de programación como Java y Python. También posee gran capacidad de almacenamiento en cuanto a la cantidad de tablas, las filas y columnas, y la base de datos en general. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez a el sistema (Guerrero, 2016).

PosgreSQL tiene además la característica de que posee PL/R como extensión de su lenguaje, lo que le permite escribir funciones de postgreSQL y funciones de agregado en el lenguaje de cálculo estadístico R.

PostgreSQL es empleado en numerosos proyectos realizados en la UCI, destacando que es el sistema gestor de bases de datos utilizado en el desarrollo del proyecto realizado para la Fiscalía General de la República. Es por tales razones que se decide emplear este SGDB en su versión 9.3.12 para el desarrollo de la herramienta propuesta.

1.9.2. Herramienta para gestionar el SGBD

pgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usan en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL. pgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita la administración. Incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad (ShareAlike, 2008).

Teniendo en cuenta las características antes mencionadas, se utiliza PgAdmin III como herramienta de diseño y administración del SGBD.

1.10. Tecnologías para el modelado

1.10.1. Lenguaje UML

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés: Unified Modeling Language), es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML tiene una notación gráfica que permite representar en mayor o menor medida todas las fases de un proyecto informático (Hernández Orallo, 2016).

Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación, etc. Para ello utiliza varios tipos diferentes de diagramas (Alegsa, 2016).

Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar (Castellano, 2016).

El lenguaje UML es aceptado por la Object Management Group (OMG) como un estándar, y actualmente casi todas las herramientas CASE (Computer Aided Software Engineering, lo que en español significa Ingeniería de Software Asistida por Computadora) y de desarrollo de software, han adoptado UML como lenguaje de modelado (Blanco, 2011).

El estándar UML 2.0 presenta un conjunto de diagramas los cuales se clasifican en estructurales y de comportamiento. Estos diagramas son de gran utilidad para trabajar en los requisitos, en el análisis del sistema, en la construcción del mismo y en su posterior despliegue.

Por las razones antes expuestas, es conveniente utilizar UML en su versión 2.0 para la modelación del proceso de sumarización lingüística y la obtención de resúmenes lingüísticos, facilitando la generación de artefactos de la metodología seleccionada.

1.10.2. Herramientas CASE

Las herramientas CASE comprenden un amplio grupo de programas que se utilizan para ayudar a las actividades del proceso del software, como el análisis de requerimientos, el modelado de sistemas, la depuración y las pruebas (Sommerville y Galipienso, 2005). Fueron desarrolladas para automatizar esos procesos y facilitar las tareas de coordinación de los eventos que necesitan ser mejorados en el ciclo de desarrollo de software (Meza, 2016).

Visual Paradigm for UML es una herramienta CASE que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, implementación y pruebas. Ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite construir diagramas de diversos tipos, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML (visual paradigm, 2016).

1.11. Conclusiones parciales del capítulo

El estudio realizado en el presente capítulo sobre el estado actual de los referentes teóricos sobre Informática Jurídica y LDS, el análisis de las tendencias actuales del uso de metodologías ágiles en el desarrollo de software, así como el empleo de herramientas para Minería de Datos, permitió llegar a las siguientes conclusiones:

- Una revisión del estado del arte de la sumarización lingüística en la actualidad, y el desarrollo de la Informática Jurídica en Cuba y el mundo, revela que en las propuestas encontradas no existen implementaciones adecuadas actualmente disponibles.
- Dentro de la sumarización lingüística de datos, el enfoque que utiliza cuantificadores difusos posee las características más adecuadas para el desarrollo de la herramienta propuesta.

Capítulo 2

Solución Propuesta

Para el desarrollo de este capítulo se presenta un nuevo componente para la obtención de resúmenes lingüísticos. Se describen los pasos a seguir para la construcción del mismo y se exponen las evidencias de la metodología de software empleada. Al finalizar, las conclusiones del capítulo.

2.1. Descripción del funcionamiento del componente

Características del componente de software

El componente que aquí se propone, encapsula las funcionalidades requeridas para desarrollar el proceso de extracción de conocimiento a partir de los datos de los Procesos Penales de la (FGR), almacenados en la base de datos. El método de descubrimiento que se implementa es la Sumarización Lingüística de Datos, particularmente el enfoque basado en la teoría de los conjuntos difusos, propuesto por (Yager, 1982) y luego desarrollada y extendida por diversos autores. Para determinar el grado de validez de los resúmenes lingüísticos se emplean las cinco medidas de calidad reportadas en (Kacprzyk y Zadrożny, 2010; Kacprzyk y Yager, 2001; Kacprzyk et al., 2000; Zadeh, 1983). Como prototipo de los resúmenes se utiliza la protoforma de tipo 0 (menos abstracta) propuesta por (Zadeh, 2002), pues se asume que se construyen previamente los cuantificadores, cualificadores y sumarizadores, para luego determinar el grado de validez del resumen.

El componente se concibe como un esquema (schema) de base de datos postgreSQL incluido en la base de datos del Sistema de Gestión Fiscal (SIGEF). Este esquema denominado “resumenes_datos” se relaciona con los esquemas “base”, “ordinario” y “sumario” de los que extrae los datos de origen (entrada) para realizar la sumarización. Para la implementación de las funcionalidades del componente se utiliza el lenguaje procedural PL/R (Conway, 2015) por lo que se adiciona a la base de datos la extención (extension) necesaria. Los resúmenes que este componente arroja como resultado, son persistidos en la tabla (table) “tb_resumenes_simples” y “tb_resumenes_complejos”. En la figura (2.1) se ofrece una vista general de la estructura del componente y su relación con los demás esquemas de la BBDD.

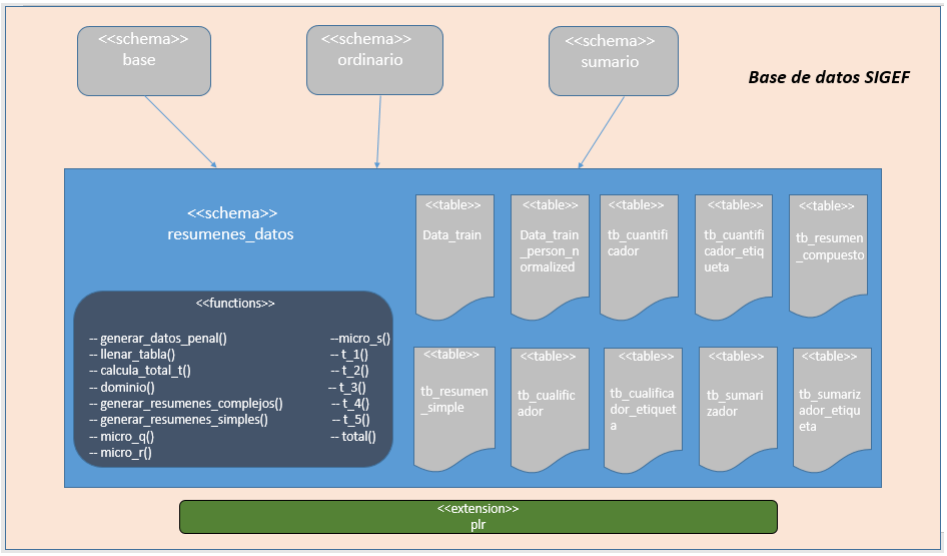


Figura 2.1: *Vista general del componente y su relación con otros esquemas.*

Para describir el funcionamiento del componente primeramente se presenta un diagrama de procesos (ver Figura 2.2) con las cuatro macro-funcionalidades principales. Posteriormente una vista secuencial de todas las funcionalidades y actividades del componente. Luego en la sección 2.2 se analizan en detalles cada uno de estos elementos.

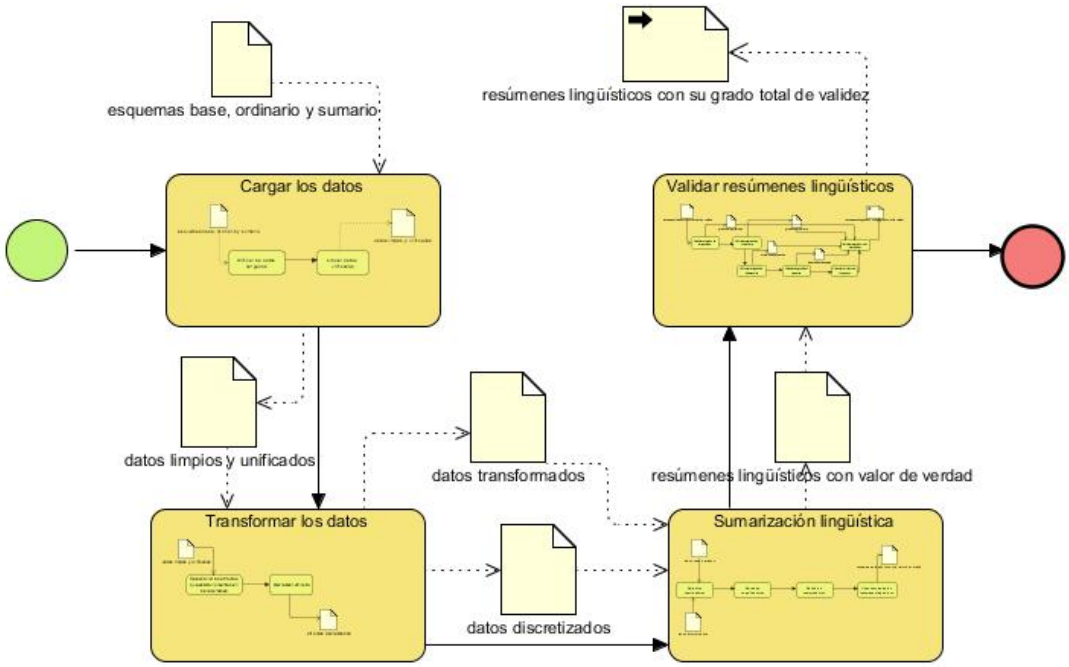


Figura 2.2: *Descripción del funcionamiento del componente.*

Vista jerárquica de las funcionalidades y actividades del componente

Funcionalidad 1: Carga de los datos

- Actividad 1.1: Unificación de los datos cargados
- Actividad 1.2: Limpieza de los datos unificados

Funcionalidad 2: Transformación de los datos

- Actividad 2.1: Selección de los atributos que estarán presentes en los resúmenes lingüísticos
- Actividad 2.2: Discretización de los datos

Funcionalidad 3: Sumarización lingüística

- Actividad 3.1: Definición de los sumarizadores
- Actividad 3.2: Definición de los cuantificadores
- Actividad 3.3: Definición de los cualificadores
- Actividad 3.4: Construcción de los resúmenes lingüísticos
- Actividad 4.1: Cálculo del valor de verdad de los resúmenes

Funcionalidad 4: Validación de los resúmenes lingüísticos

- Actividad 4.2: Cálculo del grado de impresición de los resúmenes
- Actividad 4.3: Cálculo del grado de cobertura de los resúmenes
- Actividad 4.4: Cálculo del grado de adecuación de los resúmenes
- Actividad 4.5: Calcular la longitud de los resúmenes
- Actividad 4.6: Determinar los pesos de las medidas de calidad
- Actividad 4.7: Calcular el grado total de validez

2.2. Análisis de las funcionalidades del componente propuesto

En esta sección se describen cada una de las funcionalidades descritas en la sección anterior con sus entradas y salidas y los pasos intermedios de cada funcionalidad. Se expone además la fase de KDD en la que se realiza cada actividad.

2.2.1. Análisis de la funcionalidad 1: Carga de los datos

Entrada: datos iniciales.

Para dar paso a cada una de estas actividades lo primero que se realiza es la identificación de los datos relevantes para tareas de minería de datos. Esta es una tarea que no puede ser automatizada y que debe ser realizada por el analista. Consiste en crear un conjunto de datos objetivo, seleccionando un conjunto de variables o muestra de datos, los cuales deben ser los más relevantes del proceso, así como su disponibilidad.

Unificación de los datos

Independientemente del tratamiento que se le dé a los datos, es necesario conformar un archivo único o tabla de base de datos en la que se recoja toda la información, ya que esta puede encontrarse dispersa o unificada. El objetivo de este paso intermedio, es la unifiacción de los datos con los que se va a trabajar, en una misma tabla de base de datos. En el marco de este trabajo, los datos del conjunto de entrenamiento son obtenidos de la base de datos del sistema SIGEF, como se explica en la sección anterior.

Como los datos se encuentran dispersos en varios esquemas, a través de las funciones genrar_datos_penal() y llenar_tabla() se seleccionan los atributos relevantes respecto a las materias "Ordinario" y "Sumario". Estas materias, en la base de datos, se encuentran relacionadas en los esquemas *base*, *ordinario* y *sumario*. Para almacenar estos datos, se creó la tabla *data_train*, en la cual son insertados los campos afines a los tipos de procesos contenidos en estas materias. En la figura 2.3 se muestran los atributos seleccionados.

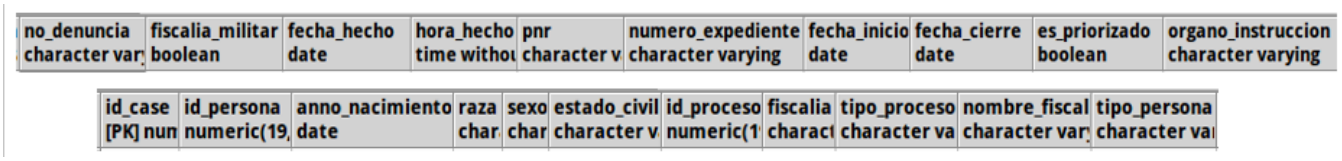


Figura 2.3: *Atributos seleccionados según la relación en los esquemas base, ordinario y sumario.*

Limpieza de los datos

El objetivo en esta actividad es crear el conjunto de datos más significativos y manejables en cuanto a tamaño. Se pretende preparar los datos para poder aplicarles los algoritmos de minería de datos. En esta fase se realizan tareas tales como la detección de valores anómalos (oultiers), valores faltantes, estudio de frecuencia, entre otros. La idea es corregir las observaciones atípicas que puedan provocar un resultado final alterado.

Para el caso de los datos faltantes (missing values) (Ramírez y Hernández, 2003), existen diferentes formas de tratarlos:

- Ignorar, siempre y cuando no sean significativos.
- Asignarle la media de los valores presentes en el atributo.

- Asignarle el valor que más predomina en el conjunto de valores del atributo de manera que no se afecte la probabilidad de ocurrencia de los valores.
- Asignarle un valor constante.

Existen algunos algoritmos que son robustos a datos faltantes, como por ejemplo los árboles de decisión (NAVARRAS, 2010). Con respecto a los datos anómalos, estos se pueden reemplazar manualmente en el caso de que no existan muchos, o automáticamente por un valor que preserve la media o la varianza en el caso de los valores numéricos; o por la moda en el caso de valores nominales. Una manera más sofisticada de estimar un valor es predecirlo a partir de los otros ejemplos (esto se llama a veces *imputación de datos perdidos*), utilizando cualquier técnica predictiva de aprendizaje automático (clasificación o regresión) (García et al., 2006). En el marco de este trabajo, el manejo de estas anomalías en los datos se hace a través de funciones que generan la moda y la media a partir de los valores del conjunto de datos de entrenamiento. Los errores encontrados en el conjunto de datos de entrenamiento son valores ausentes. A continuación se describen los campos que presentan este tipo de error.

- hora_hecho: se utiliza la media de los valores para cubrir los valores ausentes.
- fecha_hecho: se utiliza la moda de los valores para cubrir los valores ausentes.

Otro caso que debe ser tratado es la ambigüedad en los datos. Esto puede verse cuando para un mismo atributo se tienen diferentes valores para la misma observación. Para este caso se recomienda omitir una de las dos observaciones.

Como salida de esta actividad se obtiene la vista minable, donde están las diversas fuentes integradas, y los atributos relevantes seleccionados.

2.2.2. Análisis de la funcionalidad 2: Transformación de los datos

Entrada: vista minable.

Antes de realizar alguna transformación en los datos, la primera tarea es seleccionar los campos que serán parte de la vista minable. En el marco de este trabajo se conforma la vista minable, a partir de la vista obtenida en el paso anterior, con los atributos que estarán presentes en los resúmenes lingüísticos.

En este paso se realiza la modificación de los datos para generar los resúmenes lingüísticos y se seleccionan los atributos que estarán presentes en el resúmen. A continuación se describen los pasos para la transformación de los datos que se utilizan en el proceso de sumarización lingüística.

Selección de los atributos que estarán presentes en los resúmenes lingüísticos

La primera tarea a realizar es la selección de los atributos mediante los cuales se van a construir los resúmenes lingüísticos, ya que el conjunto de datos puede poseer otros atributos que no sean significativos o que no se desee que estén presentes en los resúmenes. La selección del conjunto de atributos se hace enfocado a la forma y al contenido de los resúmenes que se desean obtener de acuerdo a las necesidades e interés (Kacprzyk, 2000), apoyado en la especificación manual de un experto; por ejemplo: si se quiere

conocer la relación entre el tipo de proceso ordinario, la duración del proceso, edad de las personas, y el sexo; se podría obtener un resumen como: *La mayoría de las personas adultas del sexo masculino están involucradas en procesos ordinarios de larga duración.*

Como se menciona al inicio de la subsección, a partir de la vista obtenida en el paso anterior, se crea una vista minable con los campos o atributos que estarán presentes en los resúmenes lingüísticos. Para esto se descartan los siguientes campos:

- id_case: se trata de un identificador único, no arroja ninguna información relevante sobre los procesos penales de la Fiscalía.
- id_persona: se trata de un identificador único, no arroja ninguna información relevante sobre los procesos penales de la Fiscalía.
- no_denuncia: se trata de un identificador único, no arroja ninguna información relevante sobre los procesos penales de la Fiscalía.
- nombre_fiscal: este campo hace referencia al nombre del fiscal que atiende el proceso, no arroja ninguna información relevante sobre los procesos penales de la Fiscalía.
- tipo_persona: este campo hace referencia a si la persona es acusado, denunciante, demandado, demandante, solicitante, sancionado, etc. En el marco de este trabajo, no se considera de interés para la obtención de los resúmenes lingüísticos.
- organo_instruccion: este campo hace referencia a la entidad jurídica que se encarga de preparar el proceso. No se considera como dato relevante para los resúmenes que se desean obtener.
- fiscalia: este campo hace referencia a la Ficalía provincial o municipal que se encarga de preparar la investigación del delito. No se considera como dato relevante para los resúmenes que se desean obtener.
- pnr: este campo hace referencia a la entidad policial que ejerce su autoridad o salvaguarda el lugar del hecho y toma las medidas estipuladas ante cualquier hecho delictivo. No se considera como dato relevante para el tipo de resumen que se desea obtener.

Discretización de los atributos

Como segunda tarea para conformar la vista minable que posea los atributos que estarán presentes en los resúmenes lingüísticos, se fusionan algunos campos, los cuales conformarán un solo atributo dentro de la vista minable y/o se discretizan estos valores. A continuación se describen estos campos.

- anno_nacimiento: el valor de este campo es una fecha de la forma año-mes-dia. Este campo se transforma en el atributo edad, que aparece en la vista minable.
- fecha_inicio y fecha_cierre: estos campos se fusionan para conformar un solo atributo. Este atributo sería duracion_proceso que aparece en la vista minable.

- fecha_hecho: el valor de este campo es una fecha de la forma año-mes-dia. Este campo se transforma en el atributo momento_hecho que aparece en la vista minable.

Para llevar a cabo el proceso de discretización se propone usar los conceptos de la teoría de conjuntos difusos para manejar los valores de los atributos categóricos y cuantitativos. El enfoque basado en conjuntos difusos, permite definir fronteras difusas, lo cual aumenta la posibilidad de modelar las relaciones entre los valores de los atributos. Además, la forma en que modela la información se asemeja más a la realidad.

Es necesario definir los términos lingüísticos relacionados con cada atributo a discretizar y delimitar los intervalos de los valores numéricos para cada término. Los términos lingüísticos son el conjunto de términos que caracterizarán al atributo.

Para el conjunto de atributos seleccionados se tienen definidos los siguientes términos lingüísticos:

- edad={niño, adolescente, joven, adulto, anciano}
- duracion_proceso={corto, medio, largo}
- hora_hecho={madrugada, mañana, mediodía, tarde, noche}

A continuación se muestra el comportamiento a las variables lingüísticas edad, duracion_proceso y hora_hecho. Ver figuras 2.4, 2.5 y 2.6.

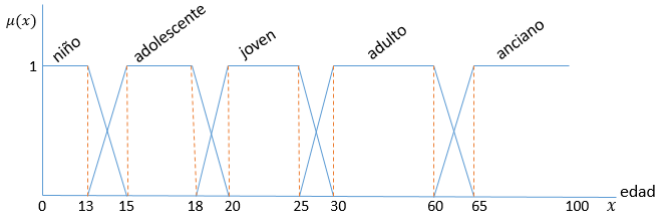


Figura 2.4: *Representación gráfica de los conjuntos difusos para la variable lingüística edad.*

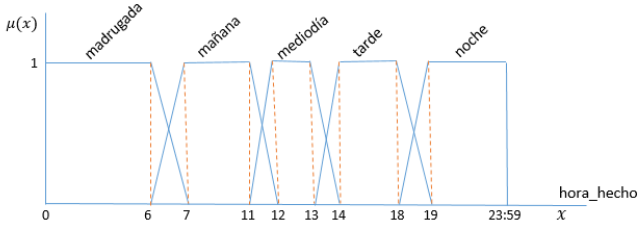


Figura 2.5: *Representación gráfica de los conjuntos difusos para la variable lingüística hora_hecho.*

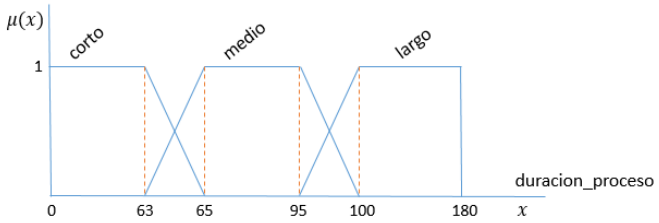


Figura 2.6: *Representación gráfica de los conjuntos difusos para la variable lingüística duracion_proceso.*

Las siguientes tablas permiten definir los intervalos para cada conjunto difuso de las variables lingüísticas edad, hora_hecho y duracion_proceso (ver figuras 2.7, 2.8 y 2.9). La figura 2.10 representa de forma genérica la función de pertenencia para un conjunto difuso.

Atributos	Corto				Medio				Largo			
	a	b	c	d	a	b	c	d	a	b	c	d
duración_proceso	0	0	60	65	63	70	90	100	95	100	180	180

Figura 2.7: *Conjuntos difusos definidos para la variable lingüística duracion_proceso.*

Atributos	Niño				Adolescente				Joven				Adulto				Anciano			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
edad	0	0	13	15	13	15	18	20	18	20	25	30	25	30	60	65	60	65	100	100

Figura 2.8: *Conjuntos difusos definidos para la variable lingüística edad.*

Atributos	Madrugada				Mañana				Mediodía				Tarde				Noche			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
hora_hecho	0	1	6	7	6	7	11	12	11	12	13	14	13	14	18	19	18	19	23:59	23:59

Figura 2.9: *Conjuntos difusos definidos para la variable lingüística hora_hecho.*

$$Trapecio \quad (x,a,b,c,d) - \begin{cases} 0 & x \le a \\ \frac{x-a}{b \quad a} & a \le x \le b \\ 1 & b \le x \le c \\ \frac{d \quad x}{d-c} & c < x < d \\ 0 & x \ge d \end{cases}$$

Figura 2.10: *Función de pertenencia de un conjunto difuso (genérica).*

Finalmente, se tiene como salida de esta actividad los datos limpios y el conjunto de atributos discretizados que se usarán para la construcción de los resúmenes lingüísticos.

2.2.3. Análisis de la funcionalidad 3: Sumarización lingüística

Entrada: vista minable con los atributos discretizados.

En este paso se definen los componentes que estarán presentes en el resúmen. Como entrada se tienen los atributos discretizados del paso anterior y finalmente se obtienen los resúmenes lingüísticos.

Definición de los sumarizadores

Como se describe en le capítulo 1, un sumarizador puede definirse como un par *atributo-valor* lingüístico, definido en el dominio del atributo. Por ejemplo, para el atributo duracion_proceso, el sumarizador puede ser duracion_proceso medio.

Luego de discretizar los datos, el par formado por el atributo y el valor lingüístico, asignado para cada observación, resultan en un sumarizador para dicho atributo. Es importante mencionar que, además de los atributos discretizados, se utilizan otros atributos en los resúmenes. Estos atributos poseen la característica de que son atributos cualitativos nominales por lo que no es necesaria su discretización; dentro de ellos podemos mencionar tipo_proceso y momento_hecho. A continuación se mencionan los sumarizadores definidos para los resúmenes.

1. momento_hecho={1 cuatrimestre, 2 cuatrimestre, 3 cuatrimestre}
2. duracion_proceso={corto, medio, largo}
3. edad={niño, adolescente, joven, adulto, anciano}
4. hora_hecho={madrugada, mañana, mediodía, tarde, noche}
5. tipo_proceso={ordinario, denuncia_atestada}. En el marco de este trabajo solo se tendrán en cuenta los tipos de proceso Ordinario y Denuncia Atestada, para la construcción de los resúmenes lingüísticos.

Definición de los cuantificadores

Los cuantificadores (Q) son una medida en la que los datos satisfacen el resumen. Estas Q se utilizan para representar la cantidad de elementos que satisfacen el predicado. Actualmente la lógica clásica se limita al uso de dos cuantificadores, existe ($\exists$) y para todo ($\forall$). Por otra parte, en el lenguaje natural se usan muchos y diversos cuantificadores, por ejemplo, la mayoría, la mitad, pocas, alrededor de 5, etc.

El uso de términos lingüísticos asociados a conjuntos difusos le dará una consistencia más humana a los resúmenes obtenidos.

Para definir correctamente los cuantificadores y obtener resúmenes con calidad que puedan ser usados fácilmente por los usuarios, para la protoforma seleccionada [1], el sistema tiene que seleccionar un cuantificador lingüístico (por lo general de un diccionario predefinido) que cuando se ponen en el lugar de Q

[1]Recordar que para la protoforma seleccionada se consideran dados S y R y se busca Q

hace la proposición lingüísticamente cuantificada resultante, válida para el más alto grado (Kacprzyk y Zadrożny, 2014).

La definición del diccionario para los cuantificadores, se obtuvo a partir del estudio de la bibliografía consultada. Este estudio permitió seleccionar dentro del conjunto de los cuantificadores absolutos, los más utilizados. Estos cuantificadores representan los extremos del dominio y no son difusos. En la figura 2.11 se muestran los cuantificadores seleccionados en el marco de este trabajo.

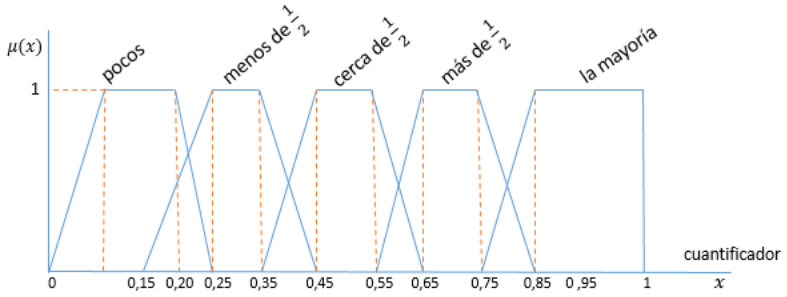


Figura 2.11: *Representación gráfica de los conjuntos difusos definidos para los cuantificadores.*

Definición de los cualificadores

De forma similar a los sumarizadores, un cualificador, es otro atributo junto con un valor lingüístico (predicado difuso) definido en el dominio del atributo A_k, como se explica al inicio del capítulo. Básicamente, los predicados de R borrosos pueden definirse haciendo una lista de sus predicados atómicos difusos (pares de *atributo/valor lingüístico*), y la estructura es como se combinan estos predicados atómicos.

Es importante mencionar que, además de los atributos discretizados, se utilizan otros atributos en los resúmenes. Estos atributos poseen la característica de que son atributos cualitativos nominales por lo que no es necesaria su discretización; dentro de ellos podemos mencionar el sexo y el estado civil de una persona. A continuación se mencionan los cualificadores seleccionados:

1. raza: toma los valores negra(n), blanca(b) o mestiza(m).

2. sexo: toma los valores masculino(m) o femenino(f).

3. estado_civil: toma los valores soltero(s), casado(c), divorciado(d) o viudo(v).

4. es_priorizado: toma los valores verdadero si el delito es priorizado, falso en caso contrario.

5. fiscalia_militar: toma los valores verdadero si el proceso es realizado por la Fiscalía Militar, falso en caso contrario.

6. edad: con los siguientes términos lingüísticos{niño, adolescente, joven, adulto, anciano}

Los sumarizadores y los cualificadores pueden ser usados indistintamente en los resúmenes lingüísticos, esto significa que, el atributo que aparece en un resumen como cualificador, puede aparecer como sumarizador en otro resumen diferente y viceversa.

Construcción de los resúmenes lingüísticos

Una vez definidos todos los elementos de la protoforma, se procede a generar los resúmenes lingüísticos. Para esto, en el caso de los resúmenes sin el cualificador:

1. Se realiza la suma del valor de la función de pertenencia μ_s desde $i = 1$ hasta n, donde n es el número de casos almacenados en la tabla de base de datos.

2. El resultado de esta suma se divide entre la cantidad de casos n (filas) almacenados en la tabla base de datos.

3. El cociente de la división obtenida en el paso anterior, es evaluado en la función de pertenencia μ_Q del cuantificador.

Como resultado final se obtienen los resúmenes lingüísticos cuantificados con su valor de verdad (T_1). Ver ecuación 1.2.

Para los resúmenes con cualificadores se realiza lo siguiente:

1. Se busca el valor de la función de pertenencia μ_s y se compara con el valor de la función de pertenencia μ_R. Con el menor valor de la comparación anterior, se realiza la suma desde $i = 1$ hasta n, donde n es el número de casos almacenados en la tabla de base de datos.

2. Luego, se busca el cociente entre el resultado de la suma anterior y la división entre n de la suma desde $i = 1$ hasta n del valor de la función μ_R .

3. El cociente obtenido en el paso anterior, se evalúa en la función de pertenencia μ_Q del cuantificador.

Como resultado final se obtienen todos los resúmenes lingüísticos cuantificados con su valor de verdad (T_1). Ver ecuación 1.3.

Finalmente se obtiene como salida de esta actividad, los resúmenes lingüísticos obtenidos.

2.2.4. Análisis de la funcionalidad 4: Validación de los resúmenes lingüísticos

Entrada: resúmenes lingüísticos

Como último paso la validación de los resúmnes lingüísticos, que tiene como entrada los resúmenes obtenidos del paso anterior. A continuación se presentan los pasos a seguir.

Cálculo de la verdad de los resúmenes lingüísticos

Como se menciona en el epígrafe 1.1.6 del capítulo 1, existen varios enfoques para calcular el grado de validez T de la proposición. Para el desarrollo de este trabajo utilizaremos los cinco criterios propuestos por (Kacprzyk y Zadrożny, 2010; Kacprzyk y Yager, 2001; Kacprzyk et al., 2000). Estos criterios son los siguientes:

1. Valor de verdad (T_1), que corresponde con lo propuesto por Zadeh, el cual se calcula mediante la ecuación 1.3 y es obtenida como resultado de la funcionalidad anterior.

2. Grado de imprecisión (T_2), que se obtiene a partir de la ecuación 1.4.

3. Grado de cobertura (T_3), que se calcula mediante la ecuación 1.6.

4. Grado de adecuación (T_4), que se obtiene con la ecuación 1.9.

5. Longitud del resumen (T_5), que es calculada por la ecuación 1.12.

Luego del cálculo de estos criterios, se determina el grado total de validez T del resumen. Una vez calculado T, la idea es encontrar un resumen óptimo S^*, el cual es el resumen de mayor grado T. El grado total de validez de un resumen, se obtiene a partir de la suma de la multiplicación de cada T_i con $i = \{1, 2, ..., 5\}$ por un peso W_j con $j = \{1, 2, ..., 5\}$, donde, la suma de los pesos es igual a 1.

El valor de los pesos puede ser predefinido o elicitado por el usuario. Otra de las formas de obtener el valor de los pesos W_j, es a través de la matriz de comparación por pares para elementos de un mismo nivel, que se utiliza en el Proceso de Análisis Jerárquico (AHP) del inglés Analytic Hierarchy Process.

El AHP es una metodología para estructurar, medir y sintetizar. Ha sido aplicado ampliamente en la solución de una gran variedad de problemas. Fue desarrollado a finales de los 60 por Thomas Saaty (Gómez y Cabrera, 2008). AHP es un método matemático creado para evaluar alternativas cuando se tienen en consideración varios criterios y está basado en el principio que la experiencia y el conocimiento de los actores son tan importantes como los datos utilizados en el proceso.

El AHP trata de desmenuzar un problema y luego unir todas las soluciones de los subproblemas en una conclusión. Hace posible la toma de decisiones grupal mediante el agregado de opiniones, de tal manera que satisfaga la relación recíproca al comparar dos elementos. Luego toma el promedio geométrico de las opiniones. Cuando el grupo consiste en expertos, cada uno elabora su propia jerarquía, y el AHP combina los resultados por el promedio geométrico (Saaty, 1990).

Finalmente se obtiene como salida de este paso, el conjunto de resúmenes lingüísticos con todos sus valores de T_i y grado total de validez.

2.3. Objeto de informatización

Mediante la realización de este trabajo, se pretende obtener un componente de sumarización lingüística que constituya una herramienta de apoyo a la toma de decisiones, basada en los procesos penales de la Fiscalía General de la República.

2.4. Fase de análisis

Dentro de la fase de análisis, la tarea más importante es la planificación. La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando "Historias de usuario", las que sustituyen a los tradicionales "casos de uso". Una vez obtenidas las historias de usuario, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

2.4.1. Historias de usuario

Las historias de usuarios sustituyen a los documentos de especificación funcional, y a los casos de uso. Estas "historias" son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. A continuación se describe 1 de las historias de usuario del sistema, el resto se encuentra en los anexos.

Tabla 2.1: Descripción de la Historia de Usuario: Seleccionar los atributos que estarán presentes en los resúmenes lingüísticos.

Historia de Usuario	
Número: 2	Nombre: Seleccionar los atributos de los cuales se quiere conocer su relación
Referencia: RF_4	
Programador: Pedro Justo Placencia Díaz	Iteración asignada: 2
Prioridad de negocio: alta	Puntos estimados: 8 días (1.3 semanas)
Riesgo de desarrollo: alta	Puntos reales: 8 días (1.3 semanas)
Descripción: Se selecciona el conjunto de atributos de los cuales se quiere conocer la relación que puede existir entre ellos.	

2.4.2. Lista de reserva del producto

Una vez definidos los requisitos del software, estos son agrupados en la lista de reserva del producto. Este artefacto permite una mayor organización logrando un mejor entendimiento por parte del equipo de desarrollo, ya que los requisitos funcionales son organizados por la prioridad que tenga cada uno y los no funcionales por su categoría. Aunque los requisitos funcionales y no funcionales no forman parte de los artefactos que se generan en la metodología XP, se considera que una descripción de estos podría facilitar el desarrollo. A continuación se muestran los requisitos definidos para la implementación de la propuesta de solución, derivados de las historias de usuario acordadas con el cliente, ver tabla 2.2 y tabla 2.3.

Tabla 2.2: Descripción de los requisitos funcionales definidos para la implementación de la solución propuesta

Requisitos Funcionales	
No.	Descripción
Cargar los datos	
RF_1	Cargar los datos.
RF_2	Unificar los datos cargados.
RF_3	Limpiar de los datos unificados.
Transformar los datos	
RF_5	Seleccionar los atributos que estarán presentes en los resúmenes lingüísticos.
RF_6	Discretizar los atributos.
RF_7	Definir los términos lingüísticos relacionados con cada atributo.
RF_8	Calcular el grado de pertenencia del atributo a cada conjunto difuso.
Sumarización Lingüística	
RF_9	Definir los sumarizadores.
RF_{10}	Definir los cuantificadores.
RF_{11}	Definir el diccionario para los cuantificadores.
RF_{12}	Definir los cualificadores.
Validar los resúmenes lingüísticos	
RF_{13}	Calcular el valor de verdad (T_1).
RF_{14}	Calcular el grado de imprecisión (T_2).
RF_{15}	Calcular el grado de cobertura (T_3).
RF_{16}	Calcular el grado de adecuación (T_4).
RF_{17}	Calcular la longitud del resumen (T_5).
RF_{18}	Obtener el resumen óptimo.

Tabla 2.3: Descripción de los requisitos no funcionales definidos para la implementación de la solución propuesta

Requisitos no Funcionales	
No.	Software
RnF_1	Para el correcto funcionamiento es necesario tener la versión 3.0.2 o superior del entorno R y la extensión del lenguaje plr para postgresql en su versión 9.3.12 o superior.
RnF_2	Se debe tener instalado el Gestor de Bases de Datos PostgreSQL en su versión 9.3.12 o superior y la herramienta PgAdmin III para la administración de la base de datos.

Continuación en la próxima página

Tabla 2.3 – *Continuación de la página anterior*

Requisitos no Funcionales	
Hardware	
RnF_3	Para garantizar un buen desempeño se necesitan computadoras con 1GB de Ram mínimo y un procesador Pentium 3.

2.4.3. Plan de iteraciones

En el plan de iteraciones, las historias de usuario seleccionadas son desarrolladas y probadas en un ciclo de iteración de acuerdo al orden prestablecido. Cada historia de usuario se traduce en tareas específicas de programación. En el plan diseñado para el desarrollo de la solución propuesta, se determinó implementar cada historia de usuario empezando por las que se encuentran dentro de la iteración uno, luego se implementan las que están enumeradas en la iteración 2 y finalmente las que pertenecen a la iteración 3. Cada historia de usuario contiene un campo con una descripción del tiempo necesario para su implementación.

Tabla 2.4: Plan de iteraciones

Iteración	Orden de la HU a implementar	Duración (días)	Duración total (semanas)
1	Cargar los datos	3	0.6
2	Seleccionar los atributos que estarán presentes en los resúmenes lingüísticos	8	1.3
2	Discretizar los atributos	7	1.2
2	Definir los cuantificadores	3	0.6
3	Sumarización Lingüística	5	1
3	Validar los resúmenes lingüísticos	10	2
Total			6.7

2.5. Fase de diseño

Para el diseño, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML. En su lugar se usan otras técnicas como las tarjetas CRC (del inglés Class-Responsibility-Collaboration). No obstante, el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante (Reyes y Ortiz, 2011).

2.5.1. Tarjetas CRC

El objetivo de las tarjetas CRC es hacer un inventario de las clases que vamos a necesitar para implementar el sistema. De esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto con el objetivo de que el diseño sea lo más simple posible verificando las especificaciones del sistema. A continuación se muestran dos de las ocho tarjetas CRC obtenidas, el resto se encuentra en los anexos.

Tabla 2.5: Descripción de la tarjeta CRC: cuantificador

Tarjeta CRC	
Clase: cuantificador	
Responsabilidad	Colaborador
Almacena todos los objetos sobre los cuales se construyen los resúmenes lingüísticos.	cuantificador_etiqueta
	sumarizador
	cualificador

Tabla 2.6: Descripción de la tarjeta CRC: resumen_simple

Tarjeta CRC	
Clase: resumen_simple	
Responsabilidad	Colaborador
Almacena los campos por los que está compuesto el resumen lingüístico sin cualificador.	

Modelo de datos

La figura 2.12 muestra el modelo de datos a través del diagrama entidad relación.

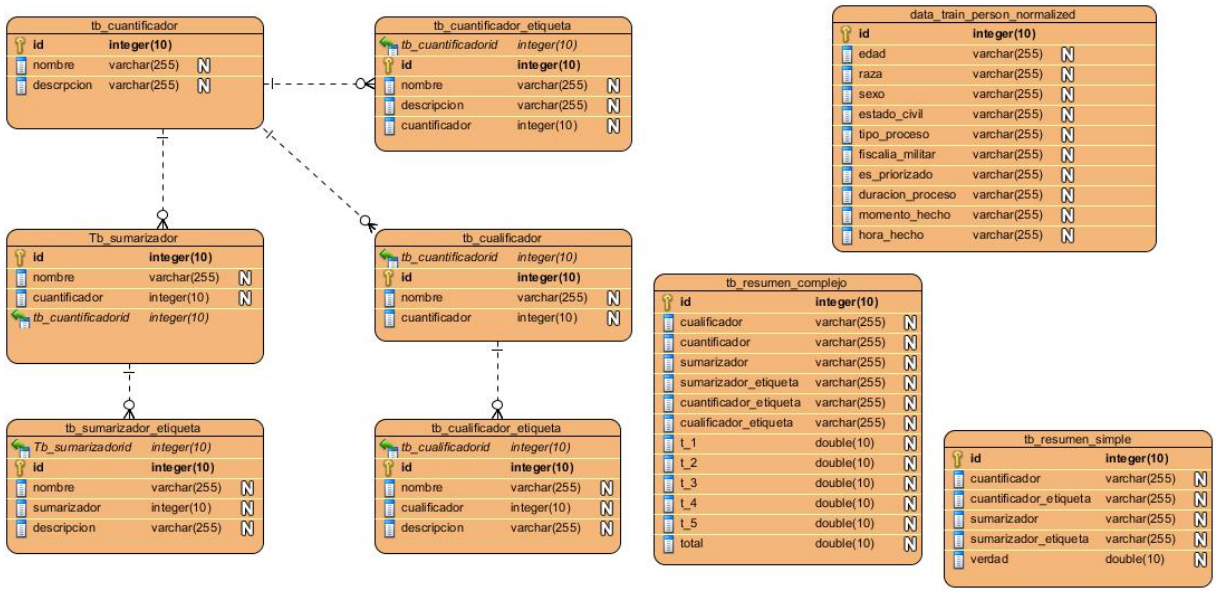


Figura 2.12: *Diagrama entidad relación para el modelo de datos.*

2.6. Fase de codificación

En esta fase se ejecutan todas las tareas de programación. Se codifica todo el diseño del sistema obteniéndose como resultado la herramienta.

2.6.1. Tareas de programación

Las tareas de programación son tarjetas de papel donde se describe que se debe realizar. Son actividades sencillas que se derivan de las historias de usuario para hacer más simple su implementación y son muy dinámicas y flexibles. Cada una de estas tares puede ser comprobada a través de los casos de prueba.

A continuación se describen las tareas de programación a desarrollar para la implemnetación del componente propuesto (tabla 2.7) y la descripción de una de ellas (ver tabla 2.8).

Tabla 2.7: Tareas de Programación o de Ingeniería

Historia de Usuario	Tareas de Programación
Cargar los datos	-Cargar los datos. -Unificar los datos. -Limpiar los datos.
Seleccionar los atributos que estarán presentes en los resúmenes lingüísticos	-Fusionar los campos fecha_inicio y fecha_cierre para obtener el nuevo atributo duracion_proceso.

Continuación en la próxima página

Tabla 2.7 – *Continuación de la página anterior*

Historia de Usuario	**Tareas de Programación**
	-Modificar el campo fecha_hecho para obtener el nuevo atributo cuatrimestre_hecho. -Modificar el campo hora_hecho. Conformar la vista minable con los atributos relevantes.
Discretizar los atributos	-Definir las etiquetas lingüísticas para cada atributo continuo. -Definir las funciones de pertenencia para cada atributo continuo.
Definir los cuantificadores	-Definir la función de pertenencia para los cuantificadores. -Construir la función plr que permita generar los cuantificadores para cada resumen.
Sumarización Lingüística	-Definir los cualificadores. -Construir la función plr que permita generar los cualificadores para cada resumen. -Construir la función plr que permita generar los sumarizadores para cada resumen. -Generar los resúmenes lingüísticos.
Validar los resúmenes lingüísticos	-Calcular valor de verdad (T_1). -Obtener grado de imprecisión (T_2). -Obtener grado de cobertura (T_3). -Calcular grado de adecuación (T_4). -Calcular longitud del resumen (T_5). -Calcular el grado total de validez T.

Tabla 2.8: Tarea de programación: Generar resúmenes lingüísticos.

Tarea de Programación	
Número Tarea: 2	Número Historia de Usuario: 5
Nombre Tarea: Generar resúmenes lingüísticos	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 5/04/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 6/04/2016	Puntos estimados: 1
Descripción: Construir la función plr que genere los resúmenes lingüísticos.	

2.6.2. Estándares de codificación

Un estándar de codificación es un conjunto de reglas de notación y nomenclatura, específicas de cada lenguaje de programación, que se usan y se siguen durante la fase de implementación de una aplicación, y reducen el riesgo de que los desarrolladores introduzcan errores que no son detectados por los compiladores, minimizando así el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismos (Charte Ojeda, 2002).

Nombres de variables, parámetros y nombre de las funciones

- Todos los nombres deben comenzar con letra minúscula.
- Los nombres que contengan varias palabras, estas deben estar separadas por un guión bajo, por ejemplo: duracion_proceso, generar_resumenes_simples.

Líneas y comentarios de código

- R no necesita punto y coma final para decir que una línea de código terminó si estas están en líneas separadas.
- Para comentar el código, se utiliza el símbolo de número (#) por ejemplo:

  # Aquí se hace ejecuta la consulta sql micro_s <- pg.spi.exec(sql)

- En la escritura de cada elemento de una línea de código, se tendrá en cuenta al menos un espacio entre cada elemento para una mayor legibilidad, por ejemplo:

  forma incorrecta: result<-0

  forma correcta: result <- 0

- Espacio de una línea entre cada línea de código para una mayor legibilidad.
- Longitud de la línea: líneas de menos de 120 caracteres.
- Rompiendo líneas: cuando una expresión tenga más de 120 caracteres, romperla después de una coma y alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior. Ver figura 2.13.

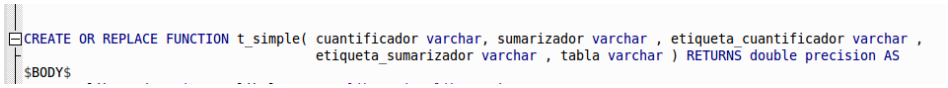


Figura 2.13: *Ruptura de línea por poseer más de 120 caracteres.*

2.6.3. Funcionalidades de R

A continuación se describen algunas de las funcionalidades de R, más utilizadas en la solución.
pg.spi.exec(): en la ejecución de una consulta SQL.
str_c(): en la construcción de cadenas de caracteres.
as.numeric(): para convertir a número.
toString(): para convertir caracteres a string.
Se utiliza además el **paquete *stringr***, el cual:

- Simplifica las operaciones de cadena mediante la eliminación de las opciones que no se necesitan.
- Produce salidas que pueden ser fácilmente utilizados como insumos. Esto incluye asegurar que las entradas que faltan dan lugar a salidas que faltan, y las entradas de longitud cero da lugar a salidas de longitud cero. También procesa factores y vectores de caracteres de la misma manera.
- Completa las funciones de manejo de cadenas de R con funciones útiles de otros lenguajes de programación.

2.6.4. Código para la obtención de los resúmenes lingüísticos

Con motivo de ejemplificar el uso de los estándares de codificación y algunas de las funcionalidades de R empleadas en el desarrollo del algoritmo, a continuación se muestra el fragmento de código que aparece en la figura 2.14, el cual responde a la generación de los resúmenes lingüísticos sin cualificador.

```
CREATE OR REPLACE FUNCTION t_simple( cuantificador varchar, sumarizador varchar , etiqueta_cuantificador varchar , etiqueta_sumarizador varchar ,
                                     tabla varchar ) RETURNS double precision AS
$BODY$
        library("stringr", lib.loc="/usr/lib/R/site-library")

        sql                     <- str_c( "SELECT ", sumarizador, " FROM ", tabla, " WHERE ", sumarizador, " <> ''" )
        rs                      <- pg.spi.exec(sql)
        n                       <- nrow(rs)
        promedio_sumarizacion   <- 0

        for ( i in 1 : n ) {

                sql                     <- str_c( "SELECT micro_s('", sumarizador, "', '" , etiqueta_sumarizador, "', '", rs[ i, 1 ], "')")
                micro_s                 <- pg.spi.exec(sql)
                promedio_sumarizacion   <- promedio_sumarizacion +  as.numeric( micro_s[ 1, 1 ] )
        }

        promedio_sumarizacion   <- promedio_sumarizacion / n;
        sql                     <- str_c( "SELECT micro_q('", cuantificador, "', '", etiqueta_cuantificador, "', ", promedio_sumarizacion, ")")
        micro_q                 <- pg.spi.exec(sql)

        return ( as.numeric( micro_q[ 1,1 ] ) )
$BODY$
LANGUAGE plr VOLATILE COST 100;
```

Figura 2.14: *Código para generar los résumenes lingüísticos simples.*

2.7. Conclusiones parciales del capítulo

El desarrollo de este capítulo, enfocado en el desarrollo de la solución propuesta y a partir de las características necesarias para cumplir el objetivo de la presente investigación, permite arribar a las siguientes

conclusiones:

- El empleo de las fases de KDD, perimitió estructurar el componente a través de funcionalidades que engloban actividades, definiendo para cada funcionalidad su correspondiente entrada y salida.
- La estructura del componente y su relación con los demás esquemas, facilitan su generalización hacia los demás procesos de la Fiscalía General de la República.
- La utilización de la matriz de comparación por pares del método AHP, permite determinar el vector de pesos de las medidas de calidad a partir del criterio de expertos con un nivel de inconsistencia inferior al umbral definido.

Capítulo 3

Fase de prueba de la Solución

En este capítulo se presenta la verificación y validación de las funcionalidades implementadas, lo cual se realiza a través de las pruebas de software definidas por la metodología. Estas pruebas tienen como objetivo comprobar que se alcanzaron los objetivos enunciados y la aceptación del cliente.

3.1. Fase de pruebas

Las pruebas son fundamentales en XP, estas constituyen una etapa dentro del desarrollo del software que permiten comprobar y revelar la calidad de un producto final. Son utilizadas para identificar fallos en la implementación o usabilidad del programa. Implican la participación directa del usuario en el desarrollo de las mismas y en la validación (Sommerville y Galipienso, 2005).

XP divide las pruebas del sistema en dos grupos: las pruebas unitarias, encargadas de verificar el código y diseñada por los programadores, y las pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (Beck y Andres, 1999).

3.1.1. Pruebas unitarias

Las pruebas unitarias se basan en realizar pruebas al código del sistema. Para llevar a cabo esta tarea se comprueban los caminos lógicos de la aplicación mediante casos de prueba, que pongan a prueba los algoritmos implementados. Las pruebas unitarias no se le pueden realizar a todo el código de la aplicación, ya que el número de caminos lógicos puede llegar a crecer de manera exponencial lo cual imposibilita realizar casos de prueba para todos estos caminos y muchos menos se podrían procesar todos. Por este motivo las pruebas de caja blanca se realizan a los principales algoritmos o procedimientos (Pressman, 2010).

Uno de los tipos de pruebas de caja blanca es la *prueba del camino básico* propuesto por (Watson et al., 1996). Esta técnica permite obtener una medida de la complejidad de un diseño procedimental o algoritmo,

y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

Para esto se representa el algoritmo o procedimiento a través de un grafo de flujo, el cual representa el flujo de control lógico del programa. En el grafo de flujo:

- Cada nodo representa una o más sentencias procedimentales.
- Un solo nodo puede corresponder a una secuencia de pasos del proceso y a una decisión.
- Las flechas (aristas) representan el flujo de control.

Cualquier representación del diseño procedimental se puede traducir a un grafo de flujo. Si en el diseño procedimental se utilizan condiciones compuestas, la generación del grafo de flujo tiene que descomponer las condiciones compuestas en condiciones sencillas.

Complejidad ciclomática

Esta métrica proporciona una medida cuantitativa de la complejidad lógica de un procedimiento. La complejidad ciclomática cuando se utiliza en el contexto de prueba del camino básico, el valor que se calcula como complejidad ciclomática, define el número de caminos independientes de un programa y nos da un límite superior para el número de casos de prueba que se deben realizar para asegurar que cada sentencia de código se ejecuta al menos una vez (Watson et al., 1996).

- La complejidad ciclomática coincide con el número de regiones del grafo de flujo. Las áreas delimitadas por aristas y nodos se denominan regiones. Cuando se contabilizan las regiones, se incluye el área exterior del grafo como otra región más.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G, se define como $V(G) = Aristas - Nodos + 2$.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G, también se define como $V(G) = nps + 1$, donde los nodos de predicado simple (nps) son aquellos de donde parten 2 o más aristas.

A partir del valor de la complejidad ciclomática obtenemos el número de caminos independientes, que nos dan un valor límite para el número de pruebas que tenemos que diseñar.

Para realizar las pruebas se seleccionó el algoritmo *t_complejo* que es una de las estructuras fundamentales entre el conjunto de funcionalidades que dan respuesta a la implementación de la herramienta para la obtención de resúmenes lingüísticos, ver figura 3.1. A continuación se describe el procedimiento para obtener los casos de prueba utilizando la técnica prueba del camino básico junto con la métrica complejidad ciclomática.

```
A___sql                    <- str_c( "SELECT ", sumarizador, ", ",cualificador, " FROM ", tabla, " WHERE ", sumarizador, " <> '' AND ",
                                     cualificador, " <> '' " )

B___rs                     <- pg.spi.exec(sql)
C___n                      <- nrow(rs)
D___promedio_sumarizacion  <- 0
E___promedio_cualificador  <- 0

F___for ( i in 1 : n ) {

G__________sql                     <- str_c( "SELECT micro_s('", sumarizador, "', '" , etiqueta_sumarizador, "', '", rs[ i, 1 ], "'),
                                             micro_r('", cualificador, "', '" , etiqueta_cualificador, "', '", rs[ i, 2 ], "')" )

H__________micro_s                 <- pg.spi.exec(sql)

I__________if( as.numeric( micro_s[ 1, 1 ] ) < as.numeric( micro_s[ 1, 2 ] ) ) {

J____________________promedio_sumarizacion   <- promedio_sumarizacion + as.numeric( micro_s[ 1, 1 ] )
           } else {

K____________________promedio_sumarizacion   <- promedio_sumarizacion + as.numeric( micro_s[ 1, 2 ] )
           }

L__________promedio_cualificador <- promedio_cualificador + as.numeric( micro_s[ 1, 2 ] )
    }

M___if( promedio_cualificador == 0 ) {

O__________promedio_cualificador = 1
    }

P___sql                    <- str_c( "SELECT micro_q('", cuantificador, "', '", etiqueta_cuantificador, "', ", as.numeric( promedio_sumarizacion / promedio_cualificador ), ")")
Q___micro_q                <- pg.spi.exec(sql)

R___return ( as.numeric( micro_q[ 1,1 ] ) )
```

Figura 3.1: *Bloque de código de la funcionalidad t_complejo el cual permite construir los resúmenes linguísticos con cualificadores.*

A continuación se muestra el grafo de flujo confeccionado a partir del código anterior. Cada nodo representa la numeración de las sentencias y los nodos de color oscuro, representan los nodos predicados del grafo, ver figura 3.2.

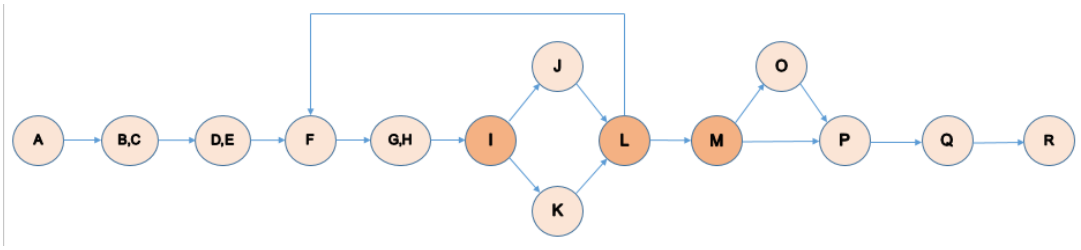


Figura 3.2: *Grafo de flujo de la funcionalidad t_complejo el cual permite construir los resúmenes lingüísticos con cualificadores.*

Complejidad ciclomática del grafo

1. $V(G) = Aristas - Nodos + 2V(G) = 16 - 14 + 2V(G) = 4$
2. $V(G) = nps + 1V(G) = 3 + 1V(G) = 4$

Una vez realizado el cálculo mediante las dos fórmulas se observa que el resultado es el mismo para ambas ecuaciones, por lo que puede afirmarse que el valor de la complejidad ciclomática es 4. Esto significa que existen 4 caminos independientes por donde el flujo puede circular y representa el límite del número de prueba a diseñar.

Los caminos independientes son:

- Camino 1: A-B-C-D-E-F-G-H-I-J-L-F-G-H-I-J-L-M-O-P-Q-R
- Camino 2: A-B-C-D-E-F-G-H-I-J-L-F-G-H-I-J-L-M-P-Q-R
- Camino 3: A-B-C-D-E-F-G-H-I-K-L-F-G-H-I-K-L-M-O-P-Q-R
- Camino 4: A-B-C-D-E-F-G-H-I-K-L-F-G-H-I-K-L-M-P-Q-R

Prueba de bucles

Los bucles son la piedra angular de la inmensa mayoría de los algoritmos implementados en software, por lo que tenemos que prestarles una atención especial a la hora de realizar la prueba del software. La prueba de bucles es una técnica de prueba de caja blanca que se centra en la validez de las construcciones de los bucles.

Se pueden definir cuatro tipos de bucles diferentes:

- bucles simples
- bucles concatenados
- bucles anidados
- bucles no estructurados

El tipo de bucle que se encuentra en el algoritmo de la figura 3.1 posee el tipo de bucle simple. A continuación se describen los tipos de prueba definidos para los bucles simples.

A los bucles simples (de n iteraciones) se les tiene que aplicar el conjunto de pruebas siguientes:

- Saltar el bucle
- Pasar sólo una vez por el bucle
- Pasar dos veces por el bucle
- Hacer m pasos del bucle con m $<$ n
- Hacer n-1, n y n+1 pasos por el bucle

Después de obtenidos los caminos básicos del flujo, se elaboran los casos de prueba para el procedimiento por cada camino básico. Para realizarlos es necesario cumplir con las siguientes especificaciones:

Descripción: se hace la entrada de datos necesaria, validando que los parámetros obligatorios no pasen vacíos al procedimiento o no se entre algún dato erróneo.

Condición de ejecución: se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.

Entrada: se muestran los parámetros que entran al procedimiento.

Resultados esperados: se expone el resultado que el procedimiento espera.

Resultados obtenidos: se muestra el resultado luego de la ejecución del camino.

Caso de prueba para el Camino 1:
Descripción: En este caso de prueba se verifica la ejecución la estructura *for* representada en el nodo F, teniendo en cuenta además el tipo de prueba definido para los bucles.

Condición de ejecución: La estructura comprobará la ejecución de las condiciones *if* y *else*. Para esto, se utilizarán 50 casos (n=50) del conjunto de entrenamiento definido en el capítulo 2 y que posteriormente serán utilizados para realizar las pruebas de aceptación.

Entrada: La base de casos que constituye los datos de entrada para la ejecución de la estructura *for* representada en el nodo F, contiene 50 tuplas.

Resultados esperados: Se espera que en la ejecución del ciclo *for*, las condicionales *if* y *else* se ejecuten cada una al menos una vez.

Resultados obtenidos: La prueba fue correcta, el ciclo se ejecuta y dentro de este las condicionales.

3.1.2. Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuario en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada.

Las pruebas de aceptación son consideradas como *pruebas de caja negra*. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información (Joskowicz, 2008).

Para realizar las pruebas de aceptación por parte del cliente, fueron elaborados dos conjuntos de entrenamiento con 387 casos de procesos Ordinario y 193 casos de Denuncia Atestada con datos reales tomados de la base de datos del sistema para la Fiscalía. Ambos conjuntos poseen un conjunto de atributos a partir de los cuales se construyen los resúmenes lingüísticos.

Conjunto 1: Proceso Ordinario

Los atributos para este conjunto son:

- edad={niño, adolescente, joven, adulto, anciano}
- duracion_proceso={corto, medio, largo}
- es_priorizado={true, false}

Conjunto 2: Denuncia Atestada
Los atributos para este conjunto son:

- hora_hecho={madrugada, mañana, mediodía, tarde, noche}
- momento_hecho={1 cuatrimestre, 2 cuatrimestre, 3 cuatrimestre}

Además de los atributos específicos para cada conjunto, se utilizan también otros atributos en la construcción de los resúmenes lingüísticos que son comunes para ambos conjuntos. Estos atributos son:

- raza: toma los valores negra(n), blanca(b) o mestiza(m).
- sexo: toma los valores masculino(m) o femenino(f).
- estado_civil: toma los valores soltero(s), casado(c), divorciado(d) o viudo(v).
- fiscalia_militar={true, false}

Procedimiento para la obtención de los pesos para el cálculo del grado total de validez de los resúmenes

Una vez definidos los conjuntos de prueba, se hace necesario definir el valor de los pesos W_i para cada T_j con i y $j = 1, 2, 3, 4, 5$ los cuales se utilizan para calcular el grado total de validez de los resúmenes. Como se mencionó anteriormente en el capítulo 2, para esto se propone usar la matriz AHP. A continuación se describe el procedimiento.

Los niveles de importancia o ponderación de los criterios se estiman por medio de comparaciones apareadas entre estos. Esta comparación se lleva a cabo usando una escala, la cual aparece en la tabla 3.1. Los valores 2, 4, 6 y 8 suelen utilizarse en situaciones intermedias, y las cifras decimales en estudios de gran precisión (Gómez y Cabrera, 2008).

Tabla 3.1: Escala fundamental para representar las intensidades de los juicios, tomado de (Saaty, 1990).

Escala numérica	Escala verbal	Explicación
1	Igual importancia.	Los dos elementos contribuyen igualmente a la propiedad o criterio.
3	Moderadamente más importante un elemento que el otro.	El juicio y la experiencia previa favorecen a un elemento frente al otro.
5	Fuertemente más importante un elemento que en otro.	El juicio y la experiencia previa favorecen fuertemente a un elemento frente al otro.
7	Mucho más fuerte la importancia de un elemento que la del otro.	Un elemento domina fuertemente. Su dominación está probada en práctica
9	Importancia extrema de un elemento frente al otro.	Un elemento domina al otro con el mayor orden de magnitud posible

Inicialmente se definen los criterios y se realiza el análisis por pares, comparando cada uno de los criterios frente a todos los demás de manera biunívoca, es decir, par a par. Los valores de cada criterio estań dados siguiendo al juicio establecido por 5 expertos, de acuerdo a la escala anteriormente expuesta. A continuación se muestra el proceso realizado por el experto 1. Las figuras que reflejan el procedimiento similar realizado por los cuatro expertos restantes, se encuentra en los anexos.

Para evaluar la selección de los pesos, se obtiene una matriz como la que se presenta en la figura 3.3. El valor 1, representa igual importancia de T_1 y T_5, el 3 indica que es moderadamente más importante T_2 que T_1, el 5 indica que es fuertemente más importante T_3 que T_1 y el 7 indica que es mucho más fuerte T_4 que T_1. La diagonal con valor 1, de color oscuro, representa la comparación del criterio con el mismo. Como la comparación se realiza en ambos sentidos, si el valor 3 indica que es moderadamente más importante T_2 que T_1, entonces la comparación de T_1 contra T_2 tiene un valor de $1/3 = 0{,}33$, y de la misma forma, ocurre para todos los demás casos.

Criterios	T1	T2	T3	T4	T5
T1	1,00	0,33	0,20	0,14	1,00
T2	3,00	1,00	0,20	0,14	3,00
T3	5,00	5,00	1,00	0,33	5,00
T4	7,00	7,00	3,00	1,00	9,00
T5	1,00	0,33	0,20	0,11	1,00
Sum	17,00	13,67	4,60	1,73	19,00

Figura 3.3: *Matriz de comparación de los criterios.*

Después de realizada las comparaciones, estas matrices son normalizadas, es decir, se divide cada término de la matriz sobre la suma de sus columnas, y en este caso se obtendría una matriz tal como se presenta en la figura 3.4. Con esta matriz, se obtiene el vector de prioridad del criterio al promediar los valores de las filas.

Matriz normalizada						Suma	Vector de Prioridad
	0,059	0,024	0,043	0,083	0,053	0,262	0,052
	0,176	0,073	0,043	0,083	0,158	0,534	0,107
	0,294	0,366	0,217	0,193	0,263	1,333	0,267
	0,412	0,512	0,652	0,578	0,474	2,628	0,526
	0,059	0,024	0,043	0,064	0,053	0,244	0,049
sum	1,000	1,000	1,000	1,000	1,000	5,000	1,000

Figura 3.4: *Matriz normalizada con vector de prioridad.*

El vector de prioridad denota la prioridad o importancia de un criterio sobre los demás. Una vez obtenido el vector de prioridad, debe calcularse el coeficiente de consistencia, el cual valida que los juicios no tengan errores entre ellos, es decir, que no se haya producido contradicciones en los mismos. Un valor de este coeficiente inferior a 10 es considerado aceptable. Para aquellos casos en que sea mayor, las opiniones y los juicios deben ser reevaluados. Esto debe aplicarse para todos los criterios de los expertos.

El coeficiente de consistencia se obtiene a través de la ecuación (CCi): CCi = CI/IA, donde, el índice de consistencia se calcula a partir de la ecuación: CI = (λ max – n) / (n-1) en la cual n, es el número de alternativas. El λ máxima se obtiene a partir de la suma de cada columna de la matriz de comparación por el vector de prioridad. Estos valores se muestran en la figura 3.5, los valores obtenidos a partir del procedimiento realizado por los cuatro expertos restantes, se encuentra en los anexos.

	lambda max	5,4102		n=	5
	Índice de consistencia (CI)	10,25%			
	Cociente de Consistencia (CR)	9,16%	**debe ser menor o igual a 10**		

Figura 3.5: *Valores de cociente de consistencia, índice de consistencia y λ máxima obtenida en el procedimiento del experto 1.*

A partir de los resultados de cada experto, se construye la matriz de comparación de criterios, donde la importancia de estos es el promedio geométrico de los valores de la matriz de comparación de crietrios obtenida de cada experto. Y como la comparación se realiza en ambos sentidos, para el caso de T_2 con respecto a T_1, el resultado asignado sería la inversa del valor de T_1 con respecto a T_2. La diagonal con valor 1, de color oscuro, es el resultado de la comparación del criterio con él mismo. Ver figura 3.6.

Matriz recíproca					
Criterios	T1	T2	T3	T4	T5
T1	1,00	0,45	0,19	0,13	1,64
T2	2,22	1,00	0,24	0,14	2,93
T3	5,19	4,23	1,00	0,30	6,12
T4	7,56	7,19	3,32	1,00	9,00
T5	0,61	0,34	0,16	0,11	1,00
Sum	16,57	13,21	4,92	1,68	20,69

Figura 3.6: *Matriz de comparación de criterios a partir del promedio geométrico.*

Una vez obtenida la matriz de comparación de criterios y de realizada las comparaciones, la matriz es normalizada, es decir, se divide cada término de la matriz sobre la suma de sus columnas, y se obtiene una matriz tal como se presenta en la figura 3.7. Con esta matriz, se obtiene el vector de prioridad del criterio al promediar los valores de las filas.

Matriz normalizada						Suma	Vector de Prioridad
	0,060	0,034	0,039	0,079	0,079	0,292	**0,058**
	0,134	0,076	0,048	0,083	0,142	0,482	**0,096**
	0,313	0,320	0,203	0,179	0,296	1,311	**0,262**
	0,456	0,544	0,676	0,594	0,435	2,705	**0,541**
	0,037	0,026	0,033	0,066	0,048	0,210	**0,042**
sum	1,000	1,000	1,000	1,000	1,000	5,000	1,000

Figura 3.7: *Matriz normalizada con vector de prioridad a partir del promedio geométrico.*

Una vez obtenido el vector de prioridad, se calcula el coeficiente de consistencia como se definió anteriormente. En la figura 3.8, se muestra el valor obtenido.

lambda max	5,3094		n=	5
Índice de consistencia (CI)	7,73%			
Cociente de Consistencia (CR)	**6,91%**	**debe ser menor o igual a 10**		

Figura 3.8: *Valores de coeficiente de consistencia, índice de consistencia y λ máxima obtenida en el procedimiento a partir del promedio geométrico.*

Una vez realizado todo el procedimiento, el valor de los pesos definidos para cada T_i con $i = \{1, 2, ..., 5\}$, es el obtenido a partir del vector de prioridad, entonces $W_1 = 0{,}058, W_2 = 0{,}096, W_3 = 0{,}262, W_4 = 0{,}541, W_5 = 0{,}042$.

Con los valores de los pesos W_i establecidos y definidos los conjuntos de entrenamiento para las pruebas de aceptación, se procede a la obtención de los resúmenes lingüísticos. A continuación se muestran imágenes de un conjunto de los resúmenes lingüísticos obtenidos, el resto se encuentra en los anexos.

Conjunto de resúmenes lingüísticos obtenidos a partir de los casos de prueba						
Resúmenes Lingüísticos	Valor de verdad (T1)	Grado de Imprecisión T(2)	Grado de cobertura (T3)	Grado de adecuación (T4)	Longitud del resumen (T5)	Grado total de validez (T)
Pocas Personas del sexo femenino están involucradas en procesos Ordinario.	1	0,5	0,17	0,16	0,12	0,31
Pocas personas ancianas están involucradas en procesos Ordinario.	1	0,78	0,16	0,16	0,003	0,35
Menos de la mitad de las personas del sexo femenino están involucradas en procesos Ordinario.	0,28	0,5	0,17	0,16	0,12	0,24
Menos de la mitad de las personas de raza negra están involucradas en procesos Ordinario.	0,55	0,60	0,39	0,39	0,06	0,41
Menos de la mitad de las personas de raza blanca están involucradas en procesos Ordinario.	0,61	0,60	0,38	0,38	0,06	0,42
Menos de la mitad de las personas de raza mestiza están involucradas en procesos Ordinario.	0,43	0,60	0,40	0,40	0,06	0,41
Menos de la mitad de las personas adultas están involucradas en procesos Ordinario.	0,61	0,78	0,39	0,39	0,01	0,45
Menos de la mitad de las personas ancianas están involucradas en procesos Ordinario.	0,20	0,78	0,16	0,16	0,01	0,27
Cerca de la mitad de las personas casadas están involucradas en procesos Ordinario.	0,55	0,68	0,40	0,40	0,03	0,43
Cerca de la mitad de las personas solteras están involucradas en procesos Ordinario.	0,75	0,68	0,42	0,42	0,03	0,47
Cerca de la mitad de las personas viudas están involucradas en procesos Ordinario.	0,38	0,68	0,38	0,38	0,03	0,41

Figura 3.9: *Conjunto de resúmenes lingüísticos obtenidos.*

3.1.3. Validación de los resultados obtenidos

La técnica Iadov constituye una vía indirecta para el estudio de la satisfacción. Mediante esta se determina el nivel de satisfacción individual y grupal a partir de una encuesta elaborada y aplicada a una muestra seleccionada. Los criterios que se utilizan están fundamentados en tres preguntas cerradas que se intercalan dentro de un cuestionario y cuya relación el sujeto desconoce. Estas tres preguntas se encuentran relacionadas a través de lo que se denomina el "El Cuadro Lógico de Iadov" (López y González, 2002), donde el número resultante de la interrelación indica la posición de cada sujeto en la escala de satisfacción, ver figura 3.10.

P3: ¿En qué medida el conocimiento que proporcionan los resúmenes lingüísticos satisface sus necesidades para la toma de decisiones?	P1: ¿Utilizaría usted los resúmenes lingüísticos que se construyen en este trabajo como conocimiento relevante en el proceso de toma de decisiones?								
	Sí			No Sé			No		
	P2: ¿Considera usted que con las funcionalidades que actualmente ofrece el SIGEF es posible obtener conocimiento relevante sobre los datos almacenados para utilizarlo como apoyo a la toma decisiones?								
	Sí	No Sé	No	Sí	No Sé	No	Sí	No Sé	No
Me satisface mucho	1	2	6	2	2	6	6	6	6
No me satisface tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
No me satisface más de lo que me satisface	6	3	6	3	4	4	3	4	4
No me satisface nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Figura 3.10: *Cuadro Lógico de Iadov.*

Para conocer el índice de satisfacción grupal acerca de la utilidad de los resúmenes lingüísticos para la

toma de decisiones, se aplica una encuesta a 15 personas, dentro de las cuales 3 son Fiscales con más de 5 años de experiencia. El resto, son Ingenieros en Ciencias Informáticas, dentro de los cuales 5 desempeñaron el rol de analista de procesos de negocio y 7 fueron miembros del equipo de desarrollo del SIGEF.

Para obtener el índice de satisfacción grupal (ISG) se parte de asociar los diferentes niveles de satisfacción individual de los encuestados en una escala numérica que oscila entre +1 y -1, donde los valores comprendidos entre -1 y -0,5 indican insatisfacción; entre -0,49 y +0,49 evidencian contradicción y entre 0,5 y 1 demuestran satisfacción. Ver figura 3.11

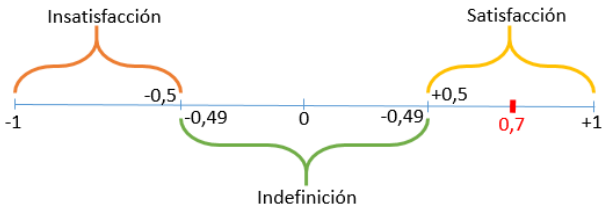


Figura 3.11: *Rangos de valoración del (ISG), (elaboración propia).*

La fórmula utilizada para obtener el índice de satisfacción grupal fue: $ISG = a(+1) + b(0{,}5) + c(0) + d(-0{,}5) + e(-1)/N$. Donde a, b, c, d, e son las cantidades de individuos de la población seleccionada, clasificados en cada una de las escalas de satisfacción respectivamente y N es la población seleccionada; siendo en este caso el ISG = 0.7. Como se puede apreciar el valor del índice es alto, lo que refleja satisfacción por parte de los encuestados, aceptación de la propuesta y reconocimiento de su utilidad.

La técnica de Iadov contempla además 3 preguntas complementarias de carácter abierto que permiten profundizar en la naturaleza de las causas que originaron los diferentes niveles de satisfacción. Resulta significativo en el análisis de estas opiniones, la preponderancia de aspectos positivos que plantearon los encuestados con respecto a la utilidad de los resúmenes lingüísticos, lo cual sirve como fundamento al valor obtenido en el ISG.

3.2. Conclusiones parciales del capítulo

Del presente capítulo se concluye que:

- Las pruebas unitarias realizadas a varios algoritmos, permiten asegurar el correcto funcionamiento de las funcionalidades implementadas.
- Se realizaron las pruebas de aceptación o funcionales, las cuales, mediante dos casos de estudio introducidos, posibilitaron la comparación positiva de los resultados en ambos casos.
- Los resultados obtenidos son validados con el empleo de la técnica de Iadov, la cual permitió definir aceptación de la propuesta y reconocimiento de su utilidad.

Conclusiones

El desarrollo de la presente investigación y los resultados generados por la misma, han permitido arribar a las siguientes conclusiones:

- En la literatura consultada, no se reportan contribuciones de aplicación de la sumarización lingüística de datos para extraer conocimiento de bases de datos de procesos penales en el mundo ni en Cuba.
- El componente desarrollado posee una estructura que permite generalizar su aplicación a otros procesos de la Fiscalía General de la República.
- La implementación del componente utilizando PL/R, permitió utilizar las potencialidades del lenguaje R para la minería de datos y facilitó su integración con la base de datos postgresql del SIGEF.
- Las pruebas unitarias y de aceptación practicadas sobre el componente resultaron satisfactorias, evidenciando el cumplimiento de los requisitos previamente definidos. Mediante la técnica de Iadov se constató un índice de satisfacción de 0.7.

De manera general se le dio cumplimiento al objetivo planteado. El componente desarrollado propone una forma fiable e interesante de conocer la relación entre varios atributos, tanto cualitativos como numéricos. Proporciona además, conocimiento nuevo en una forma entendible por el ser humano.

Recomendaciones

Para realizar sumarización lingüísticos de datos existen diferentes enfoques y todos ofrecen buenas posibilidades a la hora de encontrar información nueva en la relación que pueda existir entre los datos. Para esto es importante conocer siempre que tipo de resumen se desea obtener y principalmente el tipo de atributo con los que se trabajará. Se recomienda:

- Analizar el uso de otros enfoques de sumarización lingüísticos de datos.
- Utilizar otros atributos de la base de datos de SIGEF con el objetivo de obtener nuevo conocimiento implícito en los datos.
- Desarrollar las funcionalidades necesarias en la capa de presentación del SIGEF para facilitar el uso del componente desarrollado en este trabajo.

Bibliografía consultada y referencias bibliográficas

Leandro Alegsa. Definición de uml. 2016. URL http://www.alegsa.com.ar/Dic/uml.php.

Yarina Amoroso Fernández. La informatización del sector jurídico cubano: apuntes para una evaluación de impacto. *Revista Cubana de Derecho*, (45), 2015.

Ildar Z Batyrshin y LB Sheremetov. Perception-based approach to time series data mining. *Applied Soft Computing*, 8(3):1211–1221, 2008.

Kent Beck. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.

Kent Beck y C Andres. Extreme programming explained: Embrace chang addison-wesley professional. 1999.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, y D Thomas. Manifiesto por el desarrollo ágil de software. *Obtenido de Agile Manifesto: http://www. agilemanifesto. org/iso/es/manifesto. html*, 2001.

Yaneisy González Blanco. *Perfil de UML para los proyectos de la línea Soluciones Integrales*. Tesis Doctoral, Universidad de las Ciencias Informáticas, 2011.

Castellano. Uml. 2016. URL http://www.ecured.cu/UML#Software_libre_para_modelado_en__UML.

Rita Castillo-Ortega, Nicolás Marín, Daniel Sánchez, y Andrea GB Tettamanzi. Linguistic summarization of time series data using genetic algorithms. En *Proceedings of the 7th conference of the European Society for Fuzzy Logic and Technology*, págs. 416–423. Atlantis Press, 2011.

Francisco Charte Ojeda. Visual c# .net. *Guía práctica para usuarios, Anaya Multimedia*, 2002.

Guoqing Chen, De Liu, y Jiexun Li. Influence and conditional influence-new interestingness measures in association rule mining. En *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, tomo 3, págs. 1440–1443. IEEE, 2001.

Verónica Claramonte. Sharebi dispone de tecnologías de business intelligence para las pymes. 2012. URL http://empresayeconomia.republica.com/newsletter/sharebi-dispone-de-tecnologias-de-business-intelligence-para-las-pymes.html.

Joe Conway. Readme.plr. 2015. URL https://github.com/jconway/plr/blob/master/README.plr.

Earl Cox, Michael O'Hagan, Rodman Taber, y Michael O'Hagen. *The Fuzzy Systems Handkbook with Cdrom*. Academic Press, Inc., 1998.

Carlos Alberto Donis Díaz, Rafael Bello Pérez, y Eduardo Valencia Morales. Using linguistic data summarization in the study of creep data for the design of new steels. En *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, págs. 160–165. IEEE, 2011.

F Dıaz-Hermida y A Bugarın. Linguistic summarization of data with probabilistic fuzzy quantifiers. En *Actas del XV Congreso Espanol Sobre Tecnologıas y Lógica Fuzzy (ESTYLF 2010)*. 2010.

CA Donis-Díaz, AG Muro, R Bello-Pérez, y Eduardo Valencia Morales. A hybrid model of genetic algorithm with local search to discover linguistic data summaries from creep data. *Expert Systems with Applications*, 41(4):2035–2042, 2014.

Carlos A Donis-Diaz, Rafael Bello, y Janusz Kacprzyk. Linguistic data summarization using an enhanced genetic algorithm. *Czasopismo Techniczne*, 2015.

Carlos A Donis-Díaz, Rafael Bello, y Janusz Kacprzyk. Using ant colony optimization and genetic algorithms for the linguistic summarization of creep data. En *Intelligent Systems'2014*, págs. 81–92. Springer, 2015.

Włodzisław Duch, Rudy Setiono, et al. Computational intelligence methods for rule-based data understanding. *Proceedings of the IEEE*, 92(5):771–805, 2004.

Agustín Aguilera Estrada y Agustín Aguilera Miranda. La iuscibernetica. *Investigación Científica*, 2010.

Elio Fameli, Roberta Nannucci, y Rosa Maria Di Giorgi. Expert systems and databases: a prototype in environmental law. *Informatica e diritto*, 17(1-3):227–247, 1991.

Usama M Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, y Ramasamy Uthurusamy. Advances in knowledge discovery and data mining. 1996.

The R Foundation. What is r? 2016. URL https://www.r-project.org/about.html.

John Gantz y David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. 2012. URL https://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf.

Juan Gómez García, Javier Palarea Albaladejo, y Josep Antoni Martín Fernández. Métodos de inferencia estadística con datos faltantes: estudio de simulación sobre los efectos en las estimaciones. *Estadística española*, 48(162):241–270, 2006.

Isabel García Izquierdo. El género textual y la traducción. *Reflexiones teóricas y aplicaciones pedagógicas, Berna, Ed. Peter Lang*, 2005.

R George y R Srikanth. Data summarization using genetic algorithms and fuzzy logic. *Genetic Algorithms and Soft Computing*, págs. 599–611, 1996.

Imran Ghani. *Emerging Innovations in Agile Software Development*. addison-wesley professional, 2000.

Hilda Mar Rodríguez Gómez. Toma de decisiones. *Universidad de Costa Rica*, 2011.

Juan Carlos Osorio Gómez y Juan Pablo Orejuela Cabrera. El proceso de análisis jerárquico (ahp) y la toma de decisiones multicriterio. ejemplo de aplicación. *Scientia Et Technica*, 2(39), 2008.

Rafael Martinez Guerrero. Sobre postgresql. 2016. URL http://www.postgresql.org.es/sobre_postgresql.

Ricardo A. Guibourg. Enciclopedia de filosofÍa y teorÍa del derecho. 2015. URL http://biblio.juridicas.unam.mx/libros/libro.htm?l=3875.

Ricardo A Guibourg, Jorge O Alende, y Elena M Campanella. *Manual de informática jurídica*. Astrea, 1996.

Enrique Hernández Orallo. El lenguaje unificado de modelado (uml). 2016.

Beatriz Danmara Hidalgo García y Yasmani Otero Morfa. Mercado de datos para los departamentos de procesos penales y gestión de cuadros y personal de apoyo de la fiscalía general de la república de cuba. 2015.

Valle Joana. Informática jurídica decisional publicado en abril 12, 2011. 2011. URL http://dravallejoana.wordpress.com/2011/04/12/informatica-juridica-decisional/.

José Joskowicz. Reglas y prácticas en extreme programming. *Universidad de Vigo*, pág. 22, 2008.

J Kacprzyk y S Zadrożny. Towards human consistent data driven decision support systems using verbalization of data mining results via linguistic data summaries. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 58(3):359–370, 2010.

Janusz Kacprzyk. Studies in fuzziness and soft computing. 2000.

Janusz Kacprzyk, Anna Wilbik, y Sławomir Zadrozny. Linguistic summarization of trends: a fuzzy logic based approach. En *Proceedings of the 11th International Conference Information Processing and Management of Uncertainty in Knowledge-based Systems*, págs. 2166–2172. 2006.

Janusz Kacprzyk y Ronald R Yager. Linguistic summaries of data using fuzzy logic. *International Journal of General System*, 30(2):133–154, 2001.

Janusz Kacprzyk, Ronald R Yager, y S Zadrożny. A fuzzy logic based approach to linguistic summaries of databases. *International Journal of Applied Mathematics and Computer Science*, 10(4):813–834, 2000.

Janusz Kacprzyk y Sławomir Zadrożny. Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools. *Information Sciences*, 173(4):281–304, 2005.

Janusz Kacprzyk y Sławomir Zadrożny. Computing with words, protoforms and linguistic data summaries: towards a novel natural language based data mining and knowledge discovery tools. *Journal of Automation Mobile Robotics and Intelligent Systems*, 8, 2014.

KDNUGGETS. Analytics, data mining, data science software/tools used in the past 12 months. 2016. URL http://www.kdnuggets.com/polls/2015/analytics-data-mining-data-science-software-used.html.

María José González Labra. *Introducción a la psicología del pensamiento*. Trotta, 1998.

Ludovic Liétard. A functional interpretation of linguistic summaries of data. *Information Sciences*, 188:1–16, 2012.

Simon F LLamas. Informática jurídica. 2008. URL http://der-informatico-llamas.blogspot.com/2008/11/informtica-jurdica.html#_ftn1.

A López y V González. La técnica de iadov. una aplicación para el estudio de la satisfacción de los alumnos por las clases de educación física. *Lecturas Educación Física y Deportes, Revista Digital*, 47, 2002.

Antonio E Antonio-Enrique Pérez Luño. *Manual de informática y derecho*. 004.026/P43m. 1996.

Daimi Lamorú Marciel. *Minería de datos aplicada a expedientes ordinarios contractuales de la Sala de lo Económico del Tribunal Provincial de La Habana.* Proyecto Fin de Carrera, Instituto Superior Politécnico José Antonio Echeverría, 2015.

Goretty Carolina Martínez Bahena. La inteligencia artificial y su aplicación al campo del derecho. *Alegatos-Revista Jurídica de la Universidad Autónoma Metropolitana*, (82), 2012.

Antonio Anselmo Martino. Sistemi esperti nella giustizia. En Turín, ed., *Speciele Convegni, pag. - 38.* 1998.

YK Mathur y Abhaya Nand. Soft computing techniques and its impact in data mining. *Int. J. Emerg. Technol. Adv. Eng*, 4(8), 2014.

Tamara Benito Matías y Ma Isabel Durán Vicente. Lógica borrosa. *Universidad Carlos III*, 2008.

L Thorne McCarty. Reflections on"taxman": An experiment in artificial intelligence and legal reasoning. *Harvard Law Review*, págs. 837–893, 1977.

Jeffrey A Meldman. A preliminary study in computer-aided legal analysis. 1975.

Mirna Meza. Herramientas case. 2016. URL http://fds-herramientascase.blogspot.com/.

Juan Miguel Moine. *Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo*. Tesis Doctoral, Facultad de Informática, 2013.

NAVARRAS. Predicciones meteorológicas obtenidas de estaciones utilizando técnicas de minería de datos. 2010.

Noemí Olivera. Estado de la cuestión en la relación entre derecho e informática. *Anales de la Facultad de Ciencias Jurídicas y Sociales*, 7, 2010.

Carlos Alberto Peña Orozco. Impacto de la inteligencia artificial en el quehacer jurídico. *Revista Pensamiento Americano*, 3(5), 2013.

Monica Palmirani y Raffaella Brighi. An xml editor for legal information management. En *EGOV*, págs. 421–429. Springer, 2003.

Iván Pérez y B León. Lógica difusa para principiantes. *Publ. UCAB. Caracas*, 2007.

Eric Eduardo Piñera Trinchet. *Algoritmo de sumarización lingüística como apoyo a la toma de decisiones en gestión de proyecto*. Proyecto Fin de Carrera, Universidad de las Ciencias Informáticas, 2013.

Roger Pressman. S (2010) ingeniería del software. *Un Enfoque Práctico (7maEd.). McGraw-Hill: España*, 2010.

MJ Ramírez y J Hernández. Extracción automática de conocimiento en bases de datos e ingeniería del software. 2003.

Yunior Mesa Reyes y Yudisney Vázquez Ortiz. Espacio de comunicación e intercambio para la comunidad técnica cubana de postgresql. *Revista Cubana de Ciencias Informáticas*, 5(1), 2011.

Thomas L Saaty. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9–26, 1990.

Karla Olmos Sánchez, Jorge Rodas Osollo, y Luis Felipe Fernández. Pertinencia de la formalización de dominios semiformalmente definidos en el análisis inteligente de datos. *CULCyT*, (41), 2015.

Ken Schwaber. Agile project management with scrum. 2004.

Attribution NonCommercial ShareAlike. Pgadmin iii. 2008. URL http://www.guia-ubuntu.com/index.php?title=PgAdmin_III.

Ian Sommerville y María Isabel Alfonso Galipienso. *Ingeniería del software*. Pearson Educación, 2005.

Jennifer Stapleton. Dsdm, dynamic systems development method: The method in practice. 1997.

Julio Tellez. Derecho informático. *Editorial McGraw-Hill/Interamericana de México, SA de CV, México DF*, 1996.

Julio V Téllez. *Derecho Informático*, tomo 2. McGRAW-HILl/INTERAMERICANA DE MEXICO S.A. de c.v, 1998.

Mairelys Boeras Velázquez. Aplicando el método de boehm y turner. *Serie Científica-Universidad de las Ciencias Informáticas*, 5(6), 2012.

Ignacio Vélez Pareja. Teoria de la decisión. 2000.

visual paradigm. 2016. URL https://www.visual-paradigm.com/search/?q=visual%20paradigm%20for%20uml%208.0%20released.

Arthur H Watson, Thomas J McCabe, y Dolores R Wallace. Structured testing: A testing methodology using the cyclomatic complexity metric. *NIST special Publication*, 500(235):1–114, 1996.

Dongrui Wu y Jerry M Mendel. Linguistic summarization using if–then rules and interval type-2 fuzzy sets. *Fuzzy Systems, IEEE Transactions on*, 19(1):136–151, 2011.

Ronald R Yager. A new approach to the summarization of data. *Information Sciences*, 28(1):69–86, 1982.

Ronald R Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *Systems, Man and Cybernetics, IEEE Transactions on*, 18(1):183–190, 1988.

Lotfi A Zadeh. *The concept of a linguistic variable and its application to approximate reasoning*. Springer, 1974.

Lotfi A Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers & Mathematics with applications*, 9(1):149–184, 1983.

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, 1996.

Lotfi A Zadeh. A prototype-centered approach to adding deduction capability to search engines-the concept of protoform. En *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*, págs. 523–525. IEEE, 2002.

Apéndice A

Anexos

A.1. Historias de Usuario

Tabla A.1: Descripción de la Historia de Usuario: Validar los resúmenes lingüísticos

Historia de Usuario	
Número: 5	Nombre: Validar los resúmenes lingüísticos
Referencia: $RF_{12}, RF_{13}, RF_{14}, RF_{15}, RF_{16}, RF_{17}$	
Programador: Pedro Justo Placencia Díaz	Iteración asignada: 3
Prioridad de negocio: alta	Puntos estimados: 10 días (2 semanas)
Riesgo de desarrollo: alta	Puntos estimados: 10 días (2 semanas)
Descripción: Calcular el valor de verdad de los resúmenes lingüísticos según los 5 criterios propuestos, los cuales son: 1. Valor de verdad (T_1) 2. Grado de imprecisión (T_2) 3. Grado de cobertura (T_3) 4. Grado de adecuación (T_4) 5. Longitud del resumen (T_5) Además se calcula también el grado total de validez T a través del cual se seleccionará el resumen óptimo.	

Tabla A.2: Descripción de la Historia de Usuario: Cargar datos

Historia de Usuario	
Número: 1	Nombre: Cargar los datos
Referencia: $RF_1, RF_2, RF_3,$	
Programador: Pedro Justo Placencia Díaz	Iteración asignada: 1
Prioridad de negocio: alta	Puntos estimados: 3 días (0.6 semanas)
Riesgo de desarrollo: alta	Puntos reales: 3 días (0.6 semanas)
Descripción: Se deben cargar los datos y realizar la unificación de los mismos en una misma tabla de base de datos. Posteriormente realizar la limpieza de los datos manualmente de acuerdo al tipo de error encontrado.	

Tabla A.3: Descripción de la Historia de Usuario: Discretizar los atributos

Historia de Usuario	
Número: 3	Nombre: Discretizar los atributos
Referencia: RF_5, RF_6, RF_7	
Programador: Pedro Justo Placencia Díaz	Iteración asignada: 2
Prioridad de negocio: alta	Puntos estimados: 7 días (1.2 semana)
Riesgo de desarrollo: alta	Puntos reales: 7 días (1.2 semana)
Descripción: Mediante la teoría de conjuntos difusos se definen las etiquetas lingüísticas que caracterizarán a los atributos y se define los intervalos numéricos para cada término. Definir la función de pertenencia para cada atributo y las etiquetas lingüísticas correspondientes, definir $\mu_S(x)$ y $\mu_R(x)$.	

Tabla A.4: Descripción de la Historia de Usuario: Definir los cuantificadores

Historia de Usuario	
Número: 4	Nombre: Definir los cuantificadores
Referencia: RF_9, RF_{10}	
Programador: Pedro Justo Placencia Díaz	Iteración asignada: 2
Prioridad de negocio: alta	Puntos estimados: 3 días (0.6 semanas)
Riesgo de desarrollo: alta	Puntos estimados: 3 días (0.6 semanas)
Descripción: Definir el tipo de cuantificador (absoluto o relativo) a utilizar. Se debe definir el diccionario para los cuantificadores seleccionados y establecer el intervalo para cada uno de ellos. Definir la función de pertenencia $\mu_Q(x)$ para los cuantificadores, la cual permitirá seleccionar el cuantificador de mayor grado para el resumen.	

A.2. Tarjetas CRC

Tabla A.5: Descripción de la tarjeta CRC: cualificador

Tarjeta CRC	
Clase: cualificador	
Responsabilidad	Colaborador
Almacena los atributos que resultan en los cualificadores para los resúmenes lingüísticos.	cuantificador
	cualificador_etiqueta

Tabla A.6: Descripción de la tarjeta CRC: cuantificador_etiqueta

Tarjeta CRC	
Clase: cuantificador_etiqueta	
Responsabilidad	Colaborador
Almacena las medidas de cantidad que satisfacen los predicados de los resúmenes lingüísticos.	cuantificador

Tabla A.7: Descripción de la tarjeta CRC: sumarizador_etiqueta

Tarjeta CRC	
Clase: sumarizador_etiqueta	
Responsabilidad	Colaborador
Almacena las etiquetas definidas para cada uno de los sumarizadores los resúmenes lingüísticos.	sumarizador

Tabla A.8: Descripción de la tarjeta CRC: cualificador_etiqueta

Tarjeta CRC	
Clase: cualificador_etiqueta	
Responsabilidad	Colaborador
Almacena las etiquetas definidas para cada uno de los cualificadores de los resúmenes lingüísticos.	cualificador

Tabla A.9: Descripción de la tarjeta CRC: resumen_complejo

Tarjeta CRC	
Clase: tb_resumen_complejo	
Responsabilidad	Colaborador
Almacena todos los campos por los que está compuesto el resumen con cualificador.	

Tabla A.10: Descripción de la tarjeta CRC: sumarizador

Tarjeta CRC	
Clase: sumarizador	
Responsabilidad	Colaborador
Almacena los sumarizadores definidos para los resúmenes lingüísticos.	sumarizador_etiqueta
	cuantificador

A.3. Tareas de programación

Tabla A.11: Tarea de programación: Cargar datos.

Tarea de Programación	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Cargar los datos	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 02/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 02/03/2016	Puntos estimados: 1
Descripción: Cargar la base de datos del sistema para la Ficalía	

Tabla A.12: Tarea de programación: Unificar datos

Tarea de Programación	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Unificar los datos	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 02/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 03/03/2016	Puntos estimados: 1
Descripción: Se seleccionan los campos relevantes respecto a los procesos penales de las diferentes tablas de la base de datos de la Fiscalía y se unifican estos en una única tabla. Dicha selección es realizada por el analista y la especificación manual por un experto.	

Tabla A.13: Tarea de programación: Unificar datos

Tarea de Programación	
Número Tarea: 3	Número Historia de Usuario: 1
Nombre Tarea: Limpiar los datos	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 04/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 04/03/2016	Puntos estimados: 1
Descripción: Se realiza la limpieza de los datos de forma manual.	

Tabla A.14: Tarea de programación: Crear la vista minable.

Tarea de Programación	
Número Tarea: 2	Número Historia de Usuario: 2
Nombre Tarea: Crear la vista minable	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 16/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 16/03/2016	Puntos estimados: 1
Descripción: Conformar la vista minable con los atributos relevantes.	

Tabla A.19: Tarea de programación: Sumarización Lingüística.

Tarea de Programación	
Número Tarea: 1	Número Historia de Usuario: 5
Nombre Tarea: Sumarización Lingüística	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 31/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 4/04/2016	Puntos estimados: 1
Descripción: Definir cuales atributos serán los cualificadores y los sumarizadores. Construir la función plr que permita generar los cualificadores y los sumarizadores para cada resumen.	

Tabla A.15: Tarea de programación: Discretizar los atributos.

Tarea de Programación	
Número Tarea: 1	Número Historia de Usuario: 3
Nombre Tarea: Discretizar los atributos	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 17/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 22/03/2016	Puntos estimados: 2
Descripción: Se definen las etiquetas lingüísticas para cada atributo continuo. Construir las funciones de pertenencia para cada atributo continuo.	

Tabla A.16: Tarea de programación: Generar los sumarizadores y los cualificadores.

Tarea de Programación	
Número Tarea: 2	Número Historia de Usuario: 3
Nombre Tarea: Generar los sumarizadores y los cualificadores	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 23/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 25/03/2016	Puntos estimados: 1
Descripción: Generar las funciones plr que permitan generar los sumarizadores y los cualificadores para el resumen.	

Tabla A.17: Tarea de programación: Definir los cuantificadores.

Tarea de Programación	
Número Tarea: 1	Número Historia de Usuario: 4
Nombre Tarea: Definir los cuantificadores	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 28/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 29/03/2016	Puntos estimados: 1
Descripción: Se definen los intervalos para los cuantificadores. Definir la función de pertenencia para los cuantificadores.	

Tabla A.18: Tarea de programación: Construir la función plr para los cuantificadores.

Tarea de Programación	
Número Tarea: 2	Número Historia de Usuario: 4
Nombre Tarea: Construir la función plr para los cuantificadores	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 30/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 30/03/2016	Puntos estimados: 1
Descripción: Construir la función plr que permita generar los cuantificadores para cada uno de los resúmenes.	

Tabla A.20: Tarea de programación: Validar los resúmenes lingüísticos.

Tarea de Programación	
Número Tarea: 1	Número Historia de Usuario: 6
Nombre Tarea: Validar los resúmenes lingüísticos	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 7/04/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 18/04/2016	Puntos estimados: 2
Descripción: Calcular para cada resumen: valor de verdad (T_1) grado de imprecisión (T_2) grado de cobertura (T_3) grado de adecuación (T_4) longitud del resumen (T_5)	

Tabla A.21: Tarea de programación: Seleccionar los atributos que estarán presentes en los resúmenes lingüísticos

Tarea de Programación	
Número Tarea: 1	Número Historia de Usuario: 2
Nombre Tarea: Seleccionar los atributos que estarán presentes en los resúmenes lingüísticos	
Programador: Pedro Justo Placencia Díaz	
Fecha Inicio: 07/03/2016	Tipo de Tarea: Desarrollo
Fecha Fin: 15/03/2016	Puntos estimados: 1
Descripción: Se fusionan los campos fecha_inicio y fecha_cierre para obtener el nuevo atributo duracion_proceso. Se Modifica el campo fecha_hecho para obtener el nuevo atributo cuatrimestre_hecho y el campo hora_hecho. Se modifica el campo anno_nacimiento para obtener el atributo edad.	

A.4. Proceso de análisis jerárquico (AHP)

Matriz recíproca

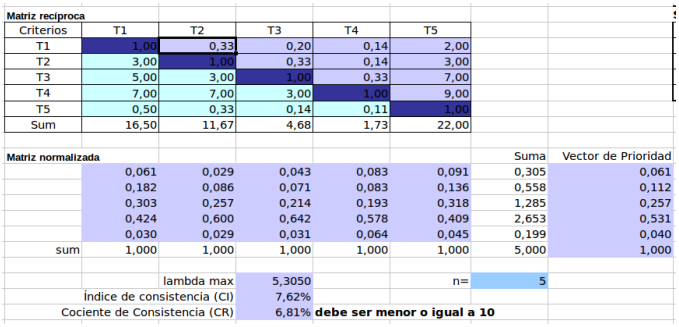
Criterios	T1	T2	T3	T4	T5
T1	1,00	0,50	0,20	0,11	1,00
T2	2,00	1,00	0,20	0,14	2,00
T3	5,00	5,00	1,00	0,20	5,00
T4	9,00	7,00	5,00	1,00	9,00
T5	1,00	0,50	0,20	0,11	1,00
Sum	18,00	14,00	6,60	1,57	18,00

Matriz normalizada

						Suma	Vector de Prioridad
	0,056	0,036	0,030	0,071	0,056	0,248	0,050
	0,111	0,071	0,030	0,091	0,111	0,415	0,083
	0,278	0,357	0,152	0,128	0,278	1,192	0,238
	0,500	0,500	0,758	0,639	0,500	2,897	0,579
	0,056	0,036	0,030	0,071	0,056	0,248	0,050
sum	1,000	1,000	1,000	1,000	1,000	5,000	1,000

lambda max	5,4292	n=	5
índice de consistencia (CI)	10,73%		
Cociente de Consistencia (CR)	9,58%	**debe ser menor o igual a 10**	

Figura A.1: *Matriz de criterios AHP del experto 2*



Matriz recíproca

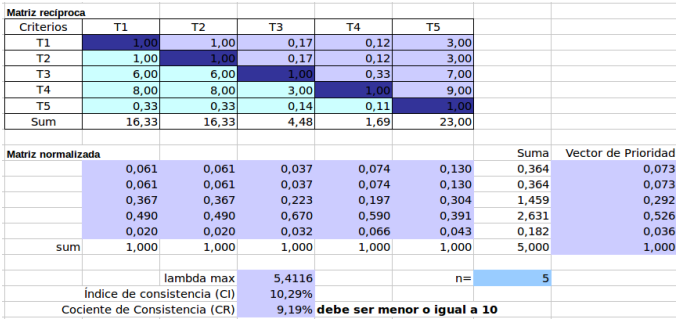
Criterios	T1	T2	T3	T4	T5
T1	1,00	0,33	0,20	0,14	2,00
T2	3,00	1,00	0,33	0,14	3,00
T3	5,00	3,00	1,00	0,33	7,00
T4	7,00	7,00	3,00	1,00	9,00
T5	0,50	0,33	0,14	0,11	1,00
Sum	16,50	11,67	4,68	1,73	22,00

Matriz normalizada

						Suma	Vector de Prioridad
	0,061	0,029	0,043	0,083	0,091	0,305	0,061
	0,182	0,086	0,071	0,083	0,136	0,558	0,112
	0,303	0,257	0,214	0,193	0,318	1,285	0,257
	0,424	0,600	0,642	0,578	0,409	2,653	0,531
	0,030	0,029	0,031	0,064	0,045	0,199	0,040
sum	1,000	1,000	1,000	1,000	1,000	5,000	1,000

lambda max	5,3050	n=	5
índice de consistencia (CI)	7,62%		
Cociente de Consistencia (CR)	6,81%	**debe ser menor o igual a 10**	

Figura A.2: *Matriz de criterios AHP del experto 3*



Matriz recíproca

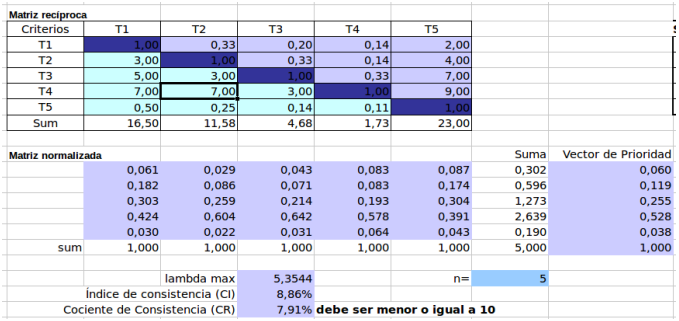
Criterios	T1	T2	T3	T4	T5
T1	1,00	1,00	0,17	0,12	3,00
T2	1,00	1,00	0,17	0,12	3,00
T3	6,00	6,00	1,00	0,33	7,00
T4	8,00	8,00	3,00	1,00	9,00
T5	0,33	0,33	0,14	0,11	1,00
Sum	16,33	16,33	4,48	1,69	23,00

Matriz normalizada

						Suma	Vector de Prioridad
	0,061	0,061	0,037	0,074	0,130	0,364	0,073
	0,061	0,061	0,037	0,074	0,130	0,364	0,073
	0,367	0,367	0,223	0,197	0,304	1,459	0,292
	0,490	0,490	0,670	0,590	0,391	2,631	0,526
	0,020	0,020	0,032	0,066	0,043	0,182	0,036
sum	1,000	1,000	1,000	1,000	1,000	5,000	1,000

lambda max	5,4116	n=	5
Índice de consistencia (CI)	10,29%		
Cociente de Consistencia (CR)	9,19%	**debe ser menor o igual a 10**	

Figura A.3: *Matriz de criterios AHP del experto 4*



Matriz recíproca

Criterios	T1	T2	T3	T4	T5
T1	1,00	0,33	0,20	0,14	2,00
T2	3,00	1,00	0,33	0,14	4,00
T3	5,00	3,00	1,00	0,33	7,00
T4	7,00	7,00	3,00	1,00	9,00
T5	0,50	0,25	0,14	0,11	1,00
Sum	16,50	11,58	4,68	1,73	23,00

Matriz normalizada

						Suma	Vector de Prioridad
	0,061	0,029	0,043	0,083	0,087	0,302	0,060
	0,182	0,086	0,071	0,083	0,174	0,596	0,119
	0,303	0,259	0,214	0,193	0,304	1,273	0,255
	0,424	0,604	0,642	0,578	0,391	2,639	0,528
	0,030	0,022	0,031	0,064	0,043	0,190	0,038
sum	1,000	1,000	1,000	1,000	1,000	5,000	1,000

lambda max	5,3544	n=	5
Índice de consistencia (CI)	8,86%		
Cociente de Consistencia (CR)	7,91%	**debe ser menor o igual a 10**	

Figura A.4: *Matriz de criterios AHP del experto 5*

A.5. Encuesta realizada

Estimado(a) colaborador, le solicitamos que responda el presente instrumento sobre la contribución de los resúmenes lingüísticos a la toma de decisiones sobre los procesos penales.

Rol ☐ Fiscal ☐ Analista de procesos de negocio ☐ Miembro equipo desarrollo SIGEF

Terminologías

Resumen lingüístico: frase en lenguaje natural, generalmente corta, que resume la esencia de un conjunto de datos (numéricos o no) que es demasiado grande para ser comprendido por el ser humano.

Algunos de los resúmenes construidos en esta investigación fueron:

Más de la mitad de los procesos de Denuncia Atestada que no son atendidos por la Fiscalía Militar ocurren en el primer cuatrimestre de año.
Todos los procesos Ordinario que no son priorizados son de corta duración
Poscas personas del sexo femenino están involucradas en procesos Ordinario.
Cerca de la mitad de los procesos de Denuncia Atestada que no son atendidos por la Fiscalía Militar ocurren en horas de la tarde.

Figura A.5: *Encuesta realizada (primera)*

Desarrollo

Responda las siguientes preguntas sobre las posibilidades de utilizar la información almacenada en la base de datos del SIGEF sobre los procesos penales.

1. ¿En qué medida el conocimiento que proporcionan los resúmenes lingüísticos satisface sus necesidades para la toma de decisiones?

( ) Me satisface mucho
( ) No me satisface tanto
( ) Me da lo mismo
( ) No me satisface más de lo que me satisface
( ) No me satisface nada
( ) No sé qué decir

2. Indique cuanto tiempo le toma extraer conocimiento relevante.

Sin disponer de los resúmenes lingüísticos	**Disponiendo** de los resúmenes lingüísticos
( ) Más de 8 horas. Tiempo aproximado: _____	( ) Más de 8 horas. Tiempo aproximado: _____
( ) Entre 3 y 8 horas. Tiempo aproximado: ____	( ) Entre 3 y 8 horas. Tiempo aproximado: _____
( ) Entre 1 y 2 horas. Tiempo aproximado: ____	( ) Entre 1 y 2 horas. Tiempo aproximado: _____

Figura A.6: *Encuesta realizada (segunda)*

3. ¿Considera usted que con las funcionalidades que actualmente ofrece el SIGEF es posible obtener conocimiento relevante sobre los datos almacenados para utilizarlo como apoyo a la toma decisiones?

( ) Si
( ) No sé
( ) No

4. ¿Considera útil la posibilidad de identificar tendencias y relaciones entre los atributos de los procesos penales como contribución a la toma de decisiones?

_____
_____
_____
_____
_____

Figura A.7: *Encuesta realizada (tercera)*

5. ¿Utilizaría usted los resúmenes lingüísticos que se construyen en este trabajo como conocimiento relevante en el proceso de toma de decisiones?

( ) Si
( ) No sé
( ) No

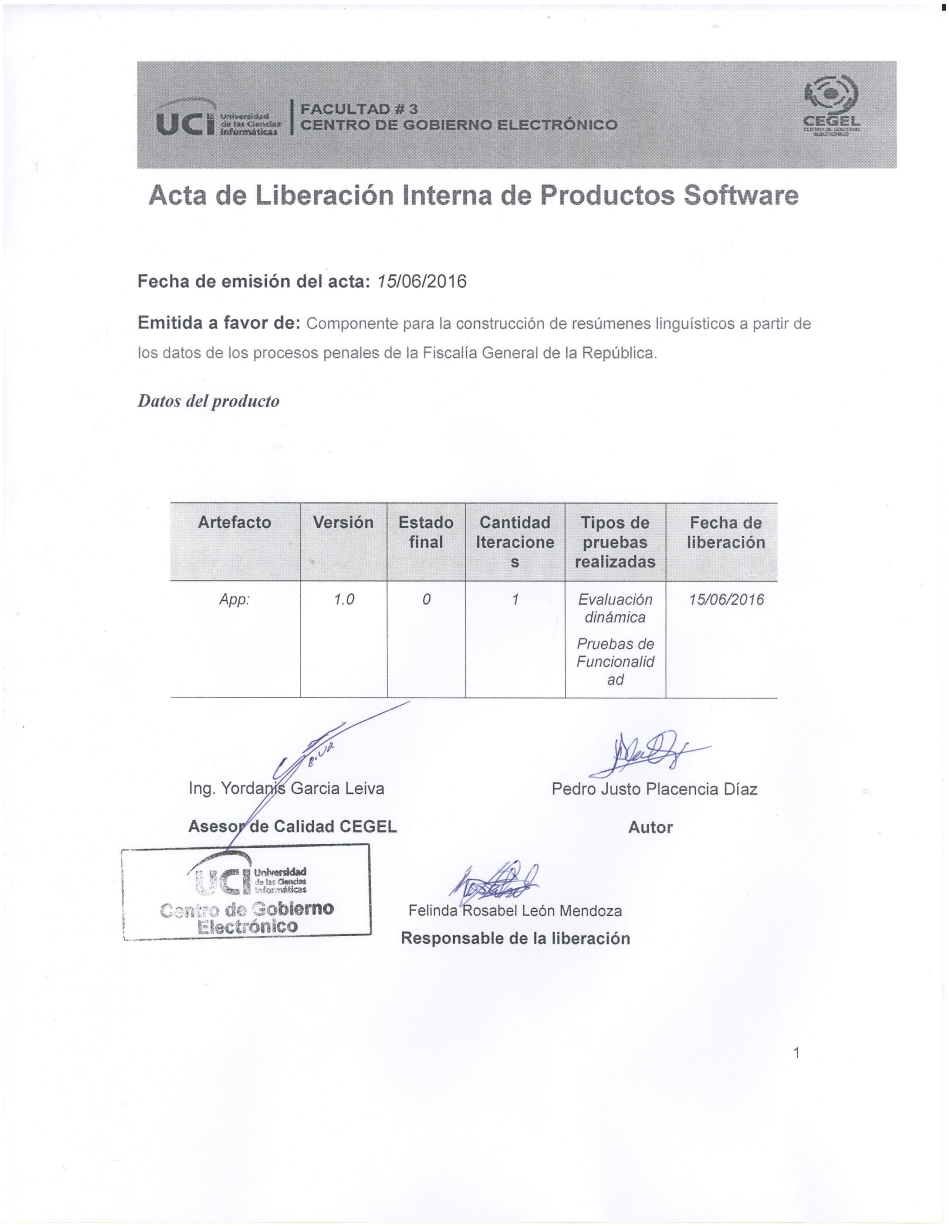
6. Según su criterio cómo describiría la compresibilidad de los resúmenes.

_____
_____
_____
_____
_____
_____

Muchas gracias por su colaboración

Figura A.8: *Encuesta realizada (cuarta)*

A.6. Acta de liberación del producto



FACULTAD # 3
CENTRO DE GOBIERNO ELECTRÓNICO

Acta de Liberación Interna de Productos Software

Fecha de emisión del acta: *15*/06/2016

Emitida a favor de: Componente para la construcción de resúmenes linguísticos a partir de los datos de los procesos penales de la Fiscalía General de la República.

Datos del producto

Artefacto	Versión	Estado final	Cantidad Iteraciones	Tipos de pruebas realizadas	Fecha de liberación
App:	*1.0*	*0*	*1*	*Evaluación dinámica* *Pruebas de Funcionalidad*	*15/06/2016*

Ing. Yordanis Garcia Leiva
Asesor de Calidad CEGEL

Pedro Justo Placencia Díaz
Autor

Universidad de las Ciencias Informáticas
Centro de Gobierno Electrónico

Felinda Rosabel León Mendoza
Responsable de la liberación

1

Figura A.9: *Acta de liberación del producto*

A.7. Resultados Obtenidos

Cerca de la mitad de las personas viudas están involucradas en procesos Ordinario.	0,38	0,68	0,38	0,38	0,03	0,41
Cerca de la mitad de las personas del sexo masculino están involucradas en procesos Ordinario.	1	0,5	0,47	0,46	0,12	0,49
Cerca de la mitad de las personas de raza negra están involucradas en procesos Ordinario.	0,44	0,60	0,39	0,39	0,06	0,40
Cerca de la mitad de las personas de raza blanca están involucradas en procesos Ordinario.	0,38	0,60	0,38	0,38	0,06	0,39
Cerca de la mitad de las personas de raza mestiza están involucradas en procesos Ordinario.	0,56	0,60	0,40	0,40	0,06	0,42
Cerca de la mitad de las personas adultas están involucradas en procesos Ordinario.	0,38	0,78	0,39	0,39	0,01	0,43
Más de la mitad de las personas adolescentes están involucradas en procesos Ordinario.	1	0,78	0,75	0,74	0,01	0,70
Ningún proceso Ordinario atendido por la Fiscalía Militar es de larga duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario no atendido por la Fiscalía Militar es de larga duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario no priorizado es de larga duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario priorizado es de larga duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario atendido por la Fiscalía Militar es de media duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario no atendido por la Fiscalía Militar es de media duración.	1	0,61	0	0	0,06	0,22

Figura A.10: *Conjunto de resúmenes lingüísticos obtenidos*

Ningún proceso Ordinario priorizado involucra adolescentes.	1	0,78	0	0,32	0,01	0,25
Ningún proceso Ordinario no priorizado involucra adolescentes.	0,84	0,78	0,02	0,02	0,01	0,25
Ningún proceso Ordinario atendido por la Fiscalía Militar involucra niños.	1	0,78	0	0	0,01	0,25
Ningún proceso Ordinario no atendido por la Fiscalía Militar involucra niños.	1	0,78	0	0	0,01	0,25
Ningún proceso Ordinario priorizado involucra niños.	1	0,78	0	0,32	0,01	0,25
Ningún proceso Ordinario no priorizado involucra niños.	1	0,78	0	0,32	0,01	0,25
Más de la mitad de los procesos Ordinario no atendidos por la Fiscalía Militar involucra adultos.	0,38	0,78	0,87	0,87	0,01	0,71
Más de la mitad de los procesos Ordinario no priorizados involucra adultos.	0,92	0,78	0,82	0,82	0,01	0,74
La mayoría de los procesos Ordinario no atendidos por la Fiscalía Militar involucra adultos.	0,61	0,78	0,87	0,87	0,01	0,74
La mayoría de los procesos Ordinario priorizados involucra adultos.	1	0,78	0,91	0,91	0,01	0,80
La mayoría de los procesos Ordinario no priorizados involucra adultos.	0,07	0,78	0,82	0,82	0,01	0,65
Todos los procesos Ordinario no atendidos por la Fiscalía Militar son de corta duración.	1	0,61	1	1	0,06	0,82
Todos los procesos Ordinario no priorizados son de corta duración.	1	0,61	1	1	0,06	0,82
Todos los procesos Ordinario priorizados son de corta duración.	1	0,61	1	1	0,06	0,82

Figura A.11: *Conjunto de resúmenes lingüísticos obtenidos*

Pocas personas de raza mestiza están involucradas en procesos de Denuncia Atestada.	1	0,60	0,16	0,16	0,06	0,32
Pocas personas adultas están involucradas en procesos de Denuncia Atestada.	1	0,78	0,18	0,18	0,01	0,37
Menos de la mitad de las personas casadas están involucradas en procesos de Denuncia Atestada.	0,43	0,68	0,19	0,19	0,03	0,29
Menos de la mitad de las personas solteras están involucradas en procesos de Denuncia Atestada.	0,65	0,68	0,21	0,21	0,03	0,33
Menos de la mitad de las personas del sexo masculino están involucradas en procesos de Denuncia Atestada.	0,47	0,5	0,19	0,18	0,1	0,27
Menos de la mitad de las personas de raza negra están involucradas en procesos de Denuncia Atestada.	0,60	0,60	0,21	0,20	0,06	0,31
Menos de la mitad de las personas de raza blanca están involucradas en procesos de Denuncia Atestada.	0,20	0,60	0,17	0,16	0,06	0,24
Menos de la mitad de las personas de raza mestiza están involucradas en procesos de Denuncia Atestada.	0,16	0,60	0,16	0,16	0,06	0,24
Menos de la mitad de las personas adultas están involucradas en procesos de Denuncia Atestada.	0,39	0,78	0,18	0,18	0,01	0,31

Figura A.12: *Conjunto de resúmenes lingüísticos obtenidos*

Menos de la mitad de los procesos de Denuncia Atestada no atendidos por la Fiscalía Militar ocurren en el tercer cuatrimestre.	0,67	0,60	0,21	0,21	0,06	0,32
Menos de la mitad de los procesos de Denuncia Atestada no atendidos por la Fiscalía Militar ocurren en la noche.	0,46	0,75	0,19	0,19	0,01	0,31
Menos de la mitad de los proceso de Denuncia Atestada que no son atendidos por la Fiscalía Militar ocurren en la tarde.	0,04	0,75	0,44	0,44	0,01	0,42
Cerca de la mitad de los procesos de Denuncia Atestada no atendidos por la Fiscalía Militar ocurren en la tarde.	0,95	0,75	0,44	0,44	0,01	0,51
Más de la mitad de los procesos de Denuncia Atestada no atendidos por la Fiscalía Militar ocurren en el primer cuatrimestre.	0,88	0,60	0,76	0,76	0,06	0,67
La mayoría de los procesos de Denuncia Atestada no atendidos por la Fiscalía Militar ocurren en el primer cuatrimestre.	0,11	0,60	0,76	0,76	0,06	0,59

Figura A.13: *Conjunto de resúmenes lingüísticos obtenidos*

Ningún proceso de Denuncia Atestada atendido por la Fiscalía Militar ocurre en el tercer cuatrimestre.	1	0,60	0	0	0,06	0,22
Ningún proceso de Denuncia Atestada atendido por la Fiscalía Militar ocurre en el segundo cuatrimestre.	1	0,60	0	0	0,06	0,22
Ningún proceso de Denuncia Atestada no atendido por la Fiscalía Militar ocurre en el segundo cuatrimestre.	0,79	0,60	0,02	0,02	0,06	0,21
Ningún proceso de Denuncia Atestada no atendido por la Fiscalía Militar ocurre en el primer cuatrimestre.	1	0,60	0	0	0,06	0,22
Ningún proceso de Denuncia Atestada atendido por la Fiscalía Militar ocurre en la noche.	1	0,75	0	0	0,01	0,25
Ningún proceso de Denuncia Atestada atendido por la Fiscalía Militar ocurre en la tarde.	1	0,75	0	0	0,01	0,25
Ningún proceso de Denuncia Atestada atendido por la Fiscalía Militar ocurre al mediodía.	1	0,75	0	0	0,01	0,25
Ningún proceso de Denuncia Atestada atendido por la Fiscalía Militar ocurre en la mañana.	1	0,75	0	0	0,01	0,25
Ningún proceso de Denuncia Atestada atendido por la Fiscalía Militar ocurre en la madrugada.	1	0,75	0	0	0,01	0,25
Ningún proceso de Denuncia Atestada no atendido por la Fiscalía Militar ocurre en la madrugada.	0,27	0,75	0,07	0,07	0,01	0,22
Pocos procesos de Denuncia Atestada no atendido por la Fiscalía Militar ocurre en el tercer cuatrimestre.	0,64	0,60	0,21	0,21	0,06	0,32
Pocos procesos de Denuncia Atestada ocurren en la noche.	1	0,75	0,19	0,19	0,01	0,37
Pocos procesos de Denuncia Atestada ocurren al mediodía.	0,59	0,75	0,12	0,12	0,01	0,29
Pocos procesos de Denuncia Atestada ocurren en la mañana.	0,69	0,75	0,13	0,13	0,01	0,30

Figura A.14: *Conjunto de resúmenes lingüísticos obtenidos.*

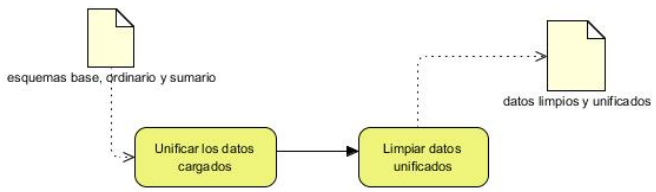
Todos los procesos Ordinario priorizados son de corta duración.	1	0,61	1	1	0,06	0,82
Ningún niño está involucrado en procesos Ordinario.	1	0,78	0	0	0,01	0,25
Ninguna persona viuda está involucrada en procesos de Denuncia Atestada.	0,10	0,68	0,08	0,08	0,03	0,20
Ninguna persona divorciada está involucrada en procesos de Denuncia Atestada.	0,41	0,68	0,05	0,05	0,03	0,21
Ningún niño está involucrado en procesos de Denuncia Atestada.	1	0,78	0	0	0,01	0,25
Ningún adolescente está involucrado en procesos de Denuncia Atestada.	1	0,78	0	0	0,01	0,25
Ninguna persona anciana está involucrada en procesos de Denuncia Atestada.	0,35	0,78	0,08	0,08	0,01	0,36
Ninguna persona soltera está involucrada en procesos de Denuncia Atestada.	0,69	0,68	0,21	0,21	0,03	0,33
Ninguna persona casada está involucrada en procesos de Denuncia Atestada.	1	0,68	0,19	0,19	0,03	0,35
Pocas personas del sexo masculino están involucradas en procesos de Denuncia Atestada.	1	0,5	0,19	0,18	0,12	0,32
Pocas personas del sexo femenino están involucradas en procesos de Denuncia Atestada.	0,44	0,5	0,12	0,10	0,12	0,22
Pocas personas de raza negra están involucradas en procesos de Denuncia Atestada.	0,79	0,60	0,21	0,20	0,06	0,33
Pocas personas de raza blanca están involucradas en procesos de Denuncia Atestada.	1	0,60	0,17	0,16	0,06	0,32

Figura A.15: *Conjunto de resúmenes lingüísticos obtenidos.*

Ningún proceso Ordinario no atendido por la Fiscalía Militar es de media duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario no priorizado es de media duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario priorizado es de media duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario atendido por la Fiscalía Militar es de corta duración.	1	0,61	0	0	0,06	0,22
Ningún proceso Ordinario atendido por la Fiscalía Militar involucra personas ancianas.	1	0,78	0	0	0,01	0,25
Ningún proceso Ordinario no atendido por la Fiscalía Militar involucra personas ancianas.	0,76	0,78	0,02	0,02	0,01	0,25
Ningún proceso Ordinario priorizado involucra personas ancianas.	0,68	0,78	0,03	0,03	0,01	0,24
Ningún proceso Ordinario no priorizado involucra personas ancianas.	0,78	0,78	0,02	0,02	0,01	0,25
Ningún proceso Ordinario atendido por la Fiscalía Militar involucra personas adultas.	1	0,78	0	0	0,01	0,25
Ningún proceso Ordinario atendido por la Fiscalía Militar involucra adolescentes.	1	0,78	0	0	0,01	0,25
Ningún proceso Ordinario no atendido por la Fiscalía Militar involucra adolescentes.	0,92	0,78	0,009	0,009	0,01	0,25

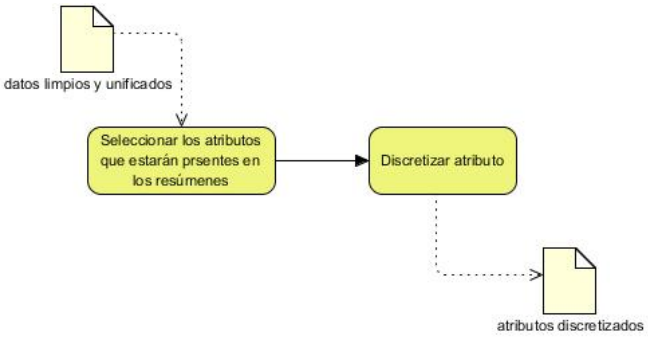
Figura A.16: *Conjunto de resúmenes lingüísticos obtenidos.*

A.8. Actividades del diagrama de procesos de negocio



los datos.jpg

Figura A.17: *Actividades del subprocesos: Cargar los datos.*



los datos.jpg

Figura A.18: *Actividades del subprocesos: Transformar los datos.*

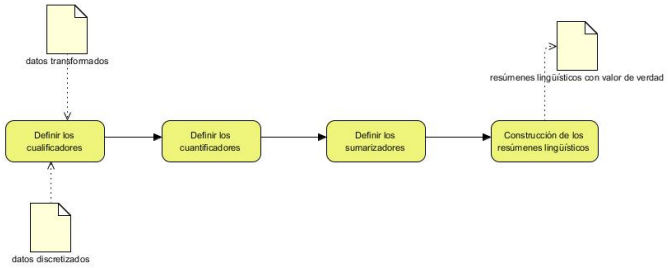
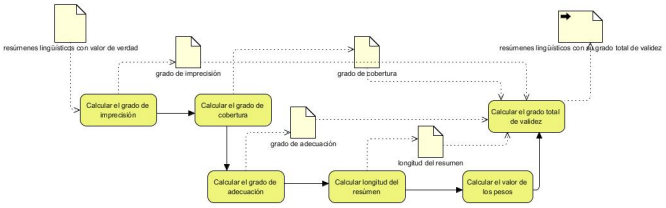


Figura A.19: *Actividades del subprocesos: Sumarización lingüística.*



resumenes.jpg

Figura A.20: *Actividades del subprocesos: Validar resúmenes lingüísticos.*