

**DIONISIO:
PROTOTIPO DE MODELADOR DE PROBLEMAS DE SIMPLEX**

**CHRISTIAN HUMBERTO CABRERA JOJOA
JAIRO ALBERTO RUANO ARISMENDY**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2011**

**DIONISIO:
PROTOTIPO DE MODELADOR DE PROBLEMAS DE SIMPLEX**

**CHRISTIAN HUMBERTO CABRERA JOJOA
JAIRO ALBERTO RUANO ARISMENDY**

**TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA
OPTAR AL TITULO DE INGENIEROS DE SISTEMAS**

**Director
ING. LUIS VICENTE CHAMORRO MARCILLO
Codirector
ING. OSCAR REVELO SANCHEZ**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2011**

NOTA DE RESPONSABILIDAD

Las ideas y conclusiones aportadas en este Trabajo de Grado son responsabilidad de los autores.

Artículo 1 del Acuerdo No. 324 de octubre 11 de 1966, emanado del honorable Concejo Directivo de la Universidad de Nariño.

NOTA DE ACEPTACIÓN

Jurado:

Jurado:

RESUMEN

En este trabajo se presenta el análisis, el diseño y la construcción de Dionisio. Un software prototipo que a partir de un ejercicio de programación lineal escrito en lenguaje natural, construye su correspondiente modelo matemático, tal como lo haría un ser humano.

El proceso de modelado de Dionisio comprende tres etapas principales: división y clasificación del texto, identificación de elementos del modelo y modelado.

Para desarrollar estas tres etapas fue necesario la aplicación de técnicas avanzadas de la computación que convergen en una serie de pasos sucesivos que imitan el proceso de modelado llevado a cabo por el ser humano, en especial se hizo uso de teorías de Inteligencia Artificial, entre los conceptos más destacados se encuentran: procesamiento del lenguaje natural, lógica difusa, reconocimiento de patrones y reglas que simulan el razonamiento humano.

ABSTRACT

This research project is about the analysis, design and construction of “Dionisio”. This is a software prototype which is based on a linear programming exercise written in natural language, and builds its own mathematical model like a human being would do it.

The modeling process “Dionisio” involves three main stages; division and classification of the text, identification of model elements and modelling.

In order to develop these three stages was necessary to apply advanced computer techniques that converge in a series of successive steps that mimic the modeling process conducted by humans. Therefore, in this project were used Artificial Intelligence theories. Thus, from those theories, there were some paramount concepts such as; natural language processing, fuzzy logic, pattern recognition and rules which simulate human reasoning.

TABLA DE CONTENIDO

1. INTRODUCCION	17
2. MARCO TEORICO	22
2.1 ANTECEDENTES	22
2.2 CONCEPTOS GENERALES	23
2.2.1 Modelado.....	23
2.2.2 Programación lineal	27
2.2.3 Inteligencia artificial.....	34
3. METODOLOGIA	43
3.1 Metodología Incremental.....	43
4. DESARROLLO DEL PROYECTO	45
4.1 TESISINTENTO1.....	45
4.1.1 Análisis.....	45
4.1.2 Diseño	48
4.2 TESISINTENTO2.....	51
4.2.1 Análisis.....	51
4.2.2 Diseño	55
4.2.3 Resultados obtenidos: Los resultados obtenidos fueron los siguientes:.....	58
4.3 TESISINTENTO3.....	60
4.3.1 Análisis.....	60
4.3.2 Diseño	64
4.3.3 Resultados obtenidos: En este intento se obtuvieron los siguientes resultados:	68
4.4 TESISINTENTO4.....	70

4.4.1 Análisis.....	70
4.4.2 Diseño	74
4.4.3 Resultados obtenidos	78
4.5 TESISINTENTO5.....	80
4.5.1 Análisis.....	80
4.5.2 Diseño	84
4.5.3 Resultados obtenidos.....	87
4.6 TESISINTENTO6 (INTENTO FINAL).....	87
4.6.1 Análisis.....	88
4.6.2 Diseño	93
4.6.4 Resultados obtenidos	106
5. DIONISIO	109
5.1 PROCESAMIENTO DE LENGUAJE NATURAL	109
5.1.1 Transformadas de Dionisio.....	110
5.1.2 Obtener objetivo.....	111
5.1.3 Separar texto.....	111
5.1.4 Clasificación de subpárrafos	112
5.2 OBTENCIÓN DE ELEMENTOS PARA EL MODELADO.....	116
5.2.1 Obtención de variables y tipos de variables	116
5.2.2 Obtención de recursos	118
5.2.3 Premodelado.....	119
5.3 MODELADO	119
5.3.1 Modelado básico	119
5.3.2 Posmodelado	120

5.4 PRESENTACIÓN DEL MODELO	121
6. TRABAJOS FUTUROS.....	123
7. CONCLUSIONES	124
8. RECOMENDACIONES.....	129
9. BIBLIOGRAFIA	130
10. CIBERGRAFIA.....	132
ANEXOS.....	133
INSTALACIÓN.....	133
INSTRUCCIONES	133
INTERFACES:	133
EJERCICIOS.....	144

TABLA DE FIGURAS

FIGURA 1 - DIAGRAMA DE CASOS DE USO GENERAL TESISINTENTO1	46
FIGURA 2 - DIAGRAMA DE CASOS DE USO TESISINTENTO1	46
FIGURA 3 - DIAGRAMA DE SECUENCIA TESISINTENTO1	47
FIGURA 4 - DIAGRAMA GENERAL DE DATOS TESISINTENTO1	48
FIGURA 5 - DIAGRAMA CONCEPTUAL TESISINTENTO1	48
FIGURA 6 - DIAGRAMA DE BASE DE DATOS TESISINTENTO1.....	49
FIGURA 7 - DIAGRAMA DE CLASES TESISINTENTO1.....	49
FIGURA 8 - DIAGRAMA DE PAQUETES TESISINTENTO1	50
FIGURA 9 - DIAGRAMA DE CASOS DE USO GENERAL TESISINTENTO2	52
FIGURA 10 - DIAGRAMA DE CASOS DE USO TESISINTENTO2	53
FIGURA 11 - DIAGRAMA DE SECUENCIA TESISINTENTO2	54
FIGURA 12 - DIAGRAMA GENERAL DE DATOS TESISINTENTO2.....	55
FIGURA 13 - DIAGRAMA CONCEPTUAL TESISINTENTO2	55
FIGURA 14 - DIAGRAMA DE BASE DE DATOS TESISINTENTO2.....	56
FIGURA 15 - DIAGRAMA DE CLASES TESISINTENTO2.....	57
FIGURA 16 - DIAGRAMA DE PAQUETES TESISINTENTO2	58
FIGURA 17 - JERARQUÍA DE CLASIFICACIÓN INICIAL.....	59
FIGURA 18 - DIAGRAMA DE CASOS DE USO GENERAL TESISINTENTO3	61
FIGURA 19 - DIAGRAMA DE CASOS DE USO TESISINTENTO3	62
FIGURA 20 - DIAGRAMA DE SECUENCIA TESISINTENTO3	63
FIGURA 21 - DIAGRAMA GENERAL DE DATOS TESISINTENTO3.....	64
FIGURA 22 - DIAGRAMA CONCEPTUAL TESISINTENTO3	65
FIGURA 23 - DIAGRAMA DE BASE DE DATOS TESISINTENTO3.....	66
FIGURA 24 - DIAGRAMA DE CLASES TESISINTENTO3.....	67
FIGURA 25 - DIAGRAMA DE PAQUETES TESISINTENTO3	68
FIGURA 26 - MATRIZ DE MODELADO INICIAL	69
FIGURA 27 - DIAGRAMA DE CASOS DE USO GENERAL TESISINTENTO4	71
FIGURA 28 - DIAGRAMA DE CASOS DE USO TESISINTENTO4	72
FIGURA 29 - DIAGRAMA DE CASOS DE USO TESISINTENTO4	73

FIGURA 30 - DIAGRAMA GENERAL DE DATOS TESISINTENTO4.....	74
FIGURA 31 - DIAGRAMA CONCEPTUAL TESISINTENTO4	75
FIGURA 32 - DIAGRAMA DE BASE DE DATOS TESISINTENTO4.....	76
FIGURA 33 - DIAGRAMA DE CLASES TESISINTENTO4.....	77
FIGURA 34 - DIAGRAMA DE PAQUETES TESISINTENTO4	78
FIGURA 35 - JERARQUÍA DE CLASIFICACIÓN FINAL.....	79
FIGURA 36 - DIAGRAMA DE CASOS DE USO GENERAL TESISINTENTO5	81
FIGURA 37 - DIAGRAMA DE CASOS DE USO TESISINTENTO5	82
FIGURA 38 - DIAGRAMA DE SECUENCIA TESISINTENTO5	83
FIGURA 39 - DIAGRAMA GENERAL DE DATOS TESISINTENTO5.....	84
FIGURA 40 - DIAGRAMA CONCEPTUAL TESISINTENTO5	84
FIGURA 41 - DIAGRAMA DE BASE DE DATOS TESISINTENTO5.....	85
FIGURA 42 - DIAGRAMA DE CLASES TESISINTENTO5.....	86
FIGURA 43 - DIAGRAMA DE PAQUETES TESISINTENTO5	87
FIGURA 44 - DIAGRAMA DE CASOS DE USO GENERAL TESISINTENTO6	89
FIGURA 45 - DIAGRAMA DE CASOS DE USO TESISINTENTO6	90
FIGURA 46 - DIAGRAMA DE SECUENCIA TESISINTENTO6	91
FIGURA 47 - DIAGRAMA GENERAL DE DATOS TESISINTENTO6.....	93
FIGURA 48 - DIAGRAMA CONCEPTUAL TESISINTENTO6	94
FIGURA 49 - DIAGRAMA DE BASE DE DATOS TESISINTENTO6.....	95
FIGURA 50 - DIAGRAMA DE CLASES TESISINTENTO6.....	96
FIGURA 51 - DIAGRAMA DE PAQUETES TESISINTENTO6	97
FIGURA 52 - PRUEBA EJERCICIO ESCRITO.....	99
FIGURA 53 - PRUEBA DIVISIÓN TEXTO	99
FIGURA 54 - PRUEBA - PRUEBA DIVISIÓN PÁRRAFO.....	100
FIGURA 55 - PRUEBA CLASIFICACIÓN SUBPÁRRAFOS.....	101
FIGURA 56 - PRUEBA OBTENCIÓN DE VARIABLES	102
FIGURA 57 - PRUEBA OBTENCIÓN DE RECURSOS.....	103
FIGURA 58 - PRUEBA UNIFICACIÓN	104
FIGURA 59 - PRUEBA ESTRUCTURA DEL MODELO	105
FIGURA 60 - PRUEBA MODELO FINAL	106
FIGURA 61 - DIAGRAMA DE ENTIDADES NIVEL 1	109
FIGURA 62 - JERARQUÍA DE CLASIFICACIÓN.....	113

FIGURA 63 - INTERFAZ PRINCIPAL.....	134
FIGURA 64 - INTERFAZ IMPORTAR TEXTO	135
FIGURA 65 - INTERFAZ CARGAR SELECCIÓN	136
FIGURA 66 - INTERFAZ EXPORTAR ARCHIVO	137
FIGURA 67 - INTERFAZ ACCESOS RÁPIDOS	139
FIGURA 68 - INTERFAZ MODELAR	140
FIGURA 69 - INTERFAZ MODELANDO TEXTO	141
FIGURA 70 - INTERFAZ PRESENTACIÓN MODELO	142
FIGURA 71 - INTERFAZ DE AYUDA	143

GLOSARIO

AMBIGUO: que puede entenderse de varios modos o admitir distintas interpretaciones.

ESTRUCTURA DE DATOS: es una forma de organizar un conjunto de datos con el objetivo de facilitar su manipulación como individualmente o como un todo.

HERENCIA: conjunto de datos transmitidos de una estructura a otra.

INVESTIGACIÓN DE OPERACIONES: es la aplicación por grupos interdisciplinarios del método científico a problemas relacionados con el control de las organizaciones o sistemas a fin de que se produzcan las soluciones que mejor sirvan a los objetivos de toda la organización.

LENGUAJE NATURAL: lenguaje utilizado normalmente por las personas.

LÉXICO: vocabulario, conjunto de palabras de una lengua, de una región, de un colectivo, una actividad, etc.

MODELO: es una representación de un objeto, sistema o idea, generalmente de forma diferente a la de la entidad misma. El propósito de los modelos es ayudar a explicar, entender o mejorar un sistema. Un modelo de un objeto puede ser una réplica exacta de éste o una abstracción de las propiedades dominantes del objeto.

MODELO MATEMÁTICO: es un tipo de modelo científico, que emplea algún tipo de formulismo matemático para expresar relaciones, proposiciones sustantivas de hechos, variables, parámetros, entidades y relaciones entre variables y/o entidades u operaciones, para estudiar comportamientos de sistemas complejos ante situaciones difíciles de observar en la realidad.

OBJETIVO DEL PROBLEMA: en programación lineal y específicamente en los problemas de Simplex, es la función que representa el deseo numérico de la entidad, mediante una combinación lineal de las variables intervinientes. Esa función puede ser para Maximizar o para Minimizar.

OPTIMIZAR: buscar la mejor manera de realizar una actividad.

PATRÓN: modelo que sirve de muestra para sacar otra cosa igual.

PESO: valor que toma un subpárrafo con respecto a un tipo de subpárrafo.

PROGRAMACIÓN LINEAL: es una de las principales ramas de la Investigación Operativa. En esta categoría se consideran todos aquellos modelos de optimización donde la función objetivo y las restricciones, son funciones lineales en las variables de decisión.

PROTOTIPO: es un ejemplar o primer molde, generalmente experimental, que será sometido a pruebas y procesos de mejoramiento con el fin de que, una vez mejorado sirva de modelo para la fabricación en serie.

SUBPÁRRAFO: segmento de texto incluido en un párrafo acotado entre una coma y una 'y' o alguna expresión de cambio de idea.

CONCEPTO DE UN LINGÜISTA HACIA DIONISIO

Por: Dr. Pedro Vicente Obando

Para contextualizar el trabajo investigativo de Dionisio debemos analizar, de manera muy general, el papel que juega el lenguaje en la historia del desarrollo humano, sus componentes esenciales y los problemas básicos que tiene el usuario de este medio de comunicación por excelencia.

El hombre, en su desarrollo de aproximadamente un millón de años, tuvo dos etapas fundamentales en la forma en la que se relacionaba con sus semejantes. En términos muy generales se podrían señalar dos momentos en el proceso de formación del lenguaje. El primer momento estaría constituido por el período en que no existía todavía lenguaje articulado en el sentido propio de la palabra; el segundo momento surge con la aparición del lenguaje articulado.

El lenguaje inarticulado fue el medio de comunicación apropiado para las formas de trabajo y la vida gregaria del hombre primitivo. El lenguaje articulado con todas sus propiedades, tanto en lo concerniente a su estructura como en lo tocante a su funcionamiento, constituye un fenómeno sumamente complejo. El lenguaje articulado resultó de la actividad del hombre primitivo aplicada al trabajo durante un largo proceso, así como del perfeccionamiento de su pensar, del prolongado desenvolvimiento de sus relaciones sociales, de su cerebro y del aparato vocal periférico.

El lenguaje articulado nace con la división del trabajo y las formas de relación social que de ello se derivan y el nivel alcanzado en la actividad pensante. La evolución del hombre llega a niveles superiores y se puede hablar del surgimiento del *Homo sapiens*. Se puede decir que la línea que divide el salvajismo del surgimiento del hombre es la aparición del lenguaje. El *Homo locuens*, quizá ya el de Cro-Magnon, es el que ha logrado la evolución y se identifica como un ser parlante capaz de comunicar sus ideas.

Esta breve síntesis del origen del lenguaje articulado lo relacionamos también con una etapa muy posterior que es el nacimiento del lenguaje escrito y el uso de diversos códigos lingüísticos que tienen una mayor relación con el trabajo adelantado en el proyecto investigativo de Dionisio. Señalemos la escritura pictográfica como esa primera etapa en la que el hombre de Neanderthal expresa formas de pensamiento en imágenes. En un salto enorme en el desarrollo de la escritura pasamos por la ideográfica o jeroglífica hasta llegar a la escritura alfabética como el grado de abstracción más elevado que le ha permitido a la sociedad humana los avances más significativos en la cultura.

El lenguaje, en todas sus manifestaciones, identifica el proceso más avanzado de la civilización. Este lenguaje moderno consta de unas partes esenciales que, sin entrar en detalles de toda su complejidad, se denominan fonemas o unidades distintivas del lenguaje, morfemas o unidades significativas del lenguaje. Con estas unidades básicas formamos palabras, frases, oraciones, discurso. Dionisio identificará texto escrito conformado por estas unidades básicas.

Dionisio como prototipo de modelador de problemas de simplex será capaz de decodificar el texto, de dividirlo en segmentos significativos como palabras, oraciones simples, oraciones compuestas. Seguramente el trabajo de Dionisio afrontará dificultades cuando tenga que interpretar códigos lingüísticos pertenecientes al lenguaje académico-formal, o quizá al lenguaje técnico, o tal vez al lenguaje popular. El lenguaje es un mundo complejo y abstracto que hemos fabricado los seres humanos.

Con todo el desarrollo de la máquina no será posible descifrar exactamente lo que aparece en el texto escrito. Muchas veces nosotros mismos no tenemos la capacidad de entender un texto complejo. El lenguaje tiene la estructura superficial que vemos en un escrito, pero también la estructura interna o inferencial que conlleva el mensaje de quien escribe un texto.

Dionisio enfrentará problemas con palabras sinónimas, homófonas que conllevarán a crear ambigüedades. Tenemos que ayudarlo con códigos ortográficos precisos, fáciles de interpretar. La redacción debe seguir el orden lógico de la oración: sujeto, verbo, objeto utilizando la puntuación adecuada que regula la estilística.

Este esfuerzo investigativo que da nacimiento a Dionisio, constituye un paso significativo en las grandes etapas del desarrollo del lenguaje, que sin duda, permitirá continuar con trabajos de investigación que poco a poco irán moviendo la barrera del conocimiento y nos acercarán al uso adecuado de estos prototipos.

Con mucha razón, al hablar de las maravillas del lenguaje y de la forma prodigiosa como adquieren los niños el dominio de una lengua, Noam Chomsky se atrevía a pensar que en esta etapa de la evolución nacemos con un chip que él denominó "Language Adquisition Device". Ese device o aparato es precisamente lo que significa Dionisio como una maravillosa máquina capaz de leer un texto.

1. INTRODUCCION

La capacidad de crear modelos que representen realidades, ha sido a lo largo de la historia una de las habilidades que distinguen al ser humano sobre las demás especies, como también una de las más difíciles de desarrollar, ya que requiere una elevada comprensión del entorno, y una gran capacidad de razonamiento por parte de la persona.

Ésta habilidad, ayuda al ser humano a entender mejor las realidades que describe, desarrollando la intuición acerca de su funcionamiento, ya que los modelos permiten establecer el comportamiento de los objetos reales, ante determinadas circunstancias controladas por el modelador.

En particular, el modelado matemático ha sido una técnica milenaria, utilizada para realizar una abstracción de una realidad que involucra relaciones numéricas, con el fin de representarla a cabalidad. Éstos modelos, son un tipo de modelo científico que utilizan fórmulas matemáticas para expresar relaciones entre hechos, variables, parámetros y entidades de una realidad, lo que permite analizar comportamientos de sistemas complejos, ante situaciones que resultan difíciles de observar, a través de la manipulación de los valores de las variables existentes, con el fin de efectuar ejercicios de precisión, explicación y previsión.

Actualmente, éstos modelos han cobrado gran importancia al ser utilizados dentro de grandes proyectos, donde la correcta planeación y planificación son claves de éxito; no obstante el proceso de construcción del modelo matemático no siempre es consistente con la realidad, lo que genera grandes pérdidas y fracasos, que se traducen en graves impactos económicos, políticos y sociales que afectan de forma directa al bienestar de las personas.

Estas inconsistencias en los modelos son producidas principalmente por errores del modelador, ya que a pesar de que la habilidad de modelar hace parte de la naturaleza humana, es difícil de desarrollar, por lo que es necesario que las instituciones académicas donde los individuos deben desarrollar esta capacidad la prioricen. Para ayudar a satisfacer esta necesidad, se plantea la creación de una herramienta que simule el proceso de modelado matemático de las personas, con el fin de que los estudiantes que están desarrollando la habilidad de modelar, puedan cotejar sus creaciones. Es necesario mencionar que, si para los seres humanos es complicado crear un modelo matemático, resulta mucho más complicado dotar a una máquina de esta habilidad. Por ello no se puede pretender que ésta herramienta tenga la capacidad de modelar cualquier problema propuesto, mejor debe ser evaluada como un primer paso en la búsqueda de una

herramienta final capaz de describir realidades en modelos matemáticos, por esta razón se limita su campo de acción a los problemas de simplex de programación lineal.

Es así como se idea Dionisio, prototipo modelador de problemas de simplex, el cual a partir del texto de un problema de programación lineal, intenta construir el modelo matemático correspondiente, para ello se necesita que Dionisio, simulé el proceso de modelado que lleva a cabo un ser humano experto en el tema, en esta simulación, Dionisio debe ser capaz de identificar los elementos del modelo matemático dentro del texto, solucionando problemas como la ambigüedad del lenguaje. A partir de estos elementos Dionisio debe armar las estructuras de los modelos como lo haría el experto, de no ser posible la construcción del modelo, Dionisio presentara las razones de su resultado y unas sugerencias para replantear el problema de una forma en la que posiblemente Dionisio construya el modelo.

A continuación, en este documento se presenta a Dionisio, el resultado de esta investigación, en primera instancia se describe la naturaleza del proyecto en cuanto a línea de investigación, alcance, definición y formulación del problema, objetivos y justificación. Posteriormente en el marco teórico se listan los posibles antecedentes de Dionisio y los conceptos utilizados en su desarrollo. Después se presenta la metodología de trabajo utilizada para luego describir el desarrollo del proyecto. En seguida se enseña como Dionisio realiza el proceso de modelado para finalmente presentar las conclusiones y recomendaciones de este estudio.

1.1 TEMA

1.1.1 Título: DIONISIO: Prototipo de Modelador de Problemas de Simplex.

1.1.2 Línea de investigación: Dionisio se encuentra inscrito bajo la línea de Optimización de Sistemas. Se enmarca dentro del área de Investigación de Operaciones y específicamente en la subárea de modelado de problemas de programación lineal.

1.1.3 Alcance y delimitación: Dionisio, contendrá el conocimiento necesario para crear modelos matemáticos de programación lineal, teniendo en cuenta las siguientes limitaciones:

- Los problemas redactados deben corresponder a problemas de Simplex, dentro del contexto de la Investigación de Operaciones con un número de variables de decisión no mayor que veinte.

- El problema estará redactado en español, con buena ortografía y puntuación, además de estar escrito con léxico de Investigación de Operaciones y, en la medida de lo posible, seguir las recomendaciones que Dionisio realiza y que serán especificadas posteriormente.
- La salida final del software será el modelo de programación lineal en lenguaje matemático correspondiente al texto redactado. Además, con un propósito académico, se listará parte del proceso que lleva a cabo Dionisio con el fin de guiar al usuario en el proceso de modelado.
- El usuario de Dionisio deberá tener conocimientos básicos de Investigación de Operaciones.

1.2. PROBLEMA OBJETO DE ESTUDIO

1.2.1 Descripción del problema: Actualmente las personas que trabajan en el área de la Investigación de Operaciones, y más particularmente sus estudiantes y profesores, no cuentan con una herramienta que les permita verificar de forma inmediata, la validez de los modelos que desarrollan. Se entiende que, el hecho de resolver mal un ejercicio académico, no trae en forma inmediata una consecuencia negativa. El verdadero problema radica en que, debido a esta falencia recurrente, las personas que supuestamente deberían estar capacitadas para modelar adecuadamente, no lo están. Como consecuencia, se encuentran grandes problemas en los procesos de modelación tanto en el nivel regional como en el nacional.

1.2.2 Formulación del problema: ¿Cómo contribuir con los usuarios del Método Simplex para que dispongan de una guía y unos resultados de comparación confiables, en el momento en que se enfrenten al modelado de problemas típicos de este método?

1.3. OBJETIVOS

1.3.1 Objetivo general: Construir Dionisio: Prototipo de Modelador de Problemas de Simplex.

1.3.2 Objetivos específicos:

- Hacer que Dionisio determine si el problema de programación lineal escrito por el usuario trata de una Maximización o de una Minimización.
- Hacer que Dionisio divida el texto escrito por el usuario en unidades de información como frases u oraciones simples.
- Hacer que Dionisio clasifique las oraciones resultantes de la división según corresponda.
- Hacer que Dionisio obtenga los elementos del modelo matemático examinando cada oración, dependiendo de su clasificación.
- Hacer que Dionisio obtenga las variables del problema.
- Hacer que Dionisio obtenga los recursos del problema.
- Hacer que Dionisio arme el modelo matemático de programación lineal a partir de los elementos obtenidos y la clasificación de las oraciones.
- Hacer que Dionisio arme la función objetivo del problema.
- Hacer que Dionisio arme un listado de restricciones del problema.
- Realizar pruebas funcionales a los diferentes módulos que componen Dionisio.
- Documentar el desarrollo del proyecto, con las temáticas y conceptos pertinentes.

1.4. JUSTIFICACION

En el estudio serio de la Investigación de Operaciones, la fase del modelado juega un papel definitivo que, desafortunadamente, los docentes y estudiantes, así como la sociedad en general, no se lo conceden.

Esta desatención puede generar graves inconsistencias en diferentes actividades que pueden conducir a millonarias pérdidas en organizaciones privadas y estatales, con la consecuente repercusión negativa en toda la comunidad.

En el campo académico, los docentes, de muy buena fe, aplican el concepto generalizado de que, “se aprende a modelar, modelando”. Su única herramienta es la formulación de problemas de grado de dificultad creciente y su único criterio de verificación es “su propio conocimiento” y “su práctica”. Ese profesional de la docencia necesita con urgencia una herramienta computacional de modelado que le permita cotejar sus creaciones.

Por su parte, el estudiante promedio modela el problema de Investigación de Operaciones; supone que ha logrado un modelo correcto, y procede a resolverlo aplicando el método matemático pertinente. Desafortunadamente, con mucha frecuencia, esa suposición es errónea. Bajo esta consideración, también al estudiante le falta una herramienta computacional que le permita verificar la validez de su creación antes de proceder a la solución matemática del modelo.

2. MARCO TEORICO

2.1 ANTECEDENTES

Efectuadas múltiples y juiciosas búsquedas, no se ha encontrado hasta el momento un antecedente directo de Dionisio. Quizás lo más aproximado es un paquete desarrollado en la Universidad Distrital Francisco José de Caldas llamado **Simplex2004**, el cual además de permitir solucionar modelos a través de métodos de Investigación de Operaciones, tiene una plantilla donde el usuario puede digitar los nombres de los elementos más importantes del problema como variables, recursos, objetivo y sus diferentes valores, para, a partir de los datos digitados, construir el modelo correspondiente. Sin embargo, a pesar de que puede ser una buena opción para enseñar a modelar, es muy limitada ya que ofrece una única plantilla para los problemas de producción dentro de una empresa.

Además, aunque **Simplex2004** ayuda a construir el modelo a partir de un texto, necesita de que el usuario identifique y digite los elementos claves dentro de la plantilla, lo que no sucede con Dionisio ya que él mismo se encarga de extraer estos elementos sin importar el tipo de problema.

Por otro lado cabe mencionar que en Investigación de Operaciones la mayoría de los esfuerzos se han centrado en programar paquetes que resuelvan modelos previamente construidos por el usuario. Tal es el caso de PHP SIMPLEX, TORA, WinQSB y SIMPLEX MEDIA. Éste último, al igual que Dionisio es un producto construido en la Universidad de Nariño.

Se han encontrado también otros paquetes computacionales denominados Lenguajes de Modelado, que son herramientas especiales para formular eficientemente modelos de programación lineal de gran tamaño. Cuando el problema presenta muchas variables o muchas restricciones, estos lenguajes facilitan las tareas de administración del modelo, tales como actualización de datos, modificación del modelo y análisis de las soluciones. Entre estos se puede destacar AMPL, MPL, GAMS y LINGO.

Se observa que, aunque estas herramientas ofrecen grandes ventajas a la hora de formular y mantener modelos a gran escala, la construcción de dichos modelos, que incluye, identificación de objetivo, determinación de variables y recursos, sigue siendo una tarea que debe ser llevada a cabo por el usuario.

Con el análisis presentado se puede afirmar, sin caer en la temeridad, que una herramienta de la naturaleza de Dionisio es única hasta el momento dadas su características de interpretación del lenguaje español escrito para generar un modelo matemático propio del Método Simplex.

2.2 CONCEPTOS GENERALES

El desarrollo de Dionisio es producto de la integración de muchos conceptos adquiridos en los cinco años de formación en el Programa de Ingeniería de Sistemas de la Universidad de Nariño, a la que se han sumado muchas horas de reflexión y pruebas que han apelado al perfil profesional del Ingeniero de Sistemas. Sin embargo cabe destacarse algunos campos del conocimiento que se definen y contextualizan a continuación, con el fin de construir la base conceptual que sostiene a Dionisio.

Se comienza por anotar que Dionisio no puede ser categorizado como un sistema especial de los que se mencionan, por lo que no debe cumplir al pie de la letra con las características y módulos que se establecen para ellos. En lugar de esto, Dionisio debe ser evaluado como una combinación de los conceptos fundamentales de cada tipo de sistema.

2.2.1 Modelado: En el lenguaje ordinario, un modelo puede entenderse como una representación de una realidad que intenta tomar los valores ideales de sus variables más importantes y dar valor cero a las variables menos importantes, haciendo que desaparezcan. De allí que el modelo sea subjetivo y cambie con los grupos sociales. En un lenguaje más técnico, un modelo es una abstracción de una realidad que involucra relaciones matemáticas y que se construye para representarla y entenderla a cabalidad. De una misma realidad pueden aparecer innumerables modelos, dependiendo del enfoque y del grado de elaboración. Un modelo no puede ser erróneo o equivocado, pero sí puede ser desenfocado.

Los modelos ayudan a entender mejor las realidades que describen, desarrollando la intuición acerca de su funcionamiento. El modelo permite interpolar y extrapolar valores de las variables integrantes, con el fin de efectuar ejercicios de precisión, explicación y previsión. En la actualidad se resalta su utilidad en los trabajos de “what if”, ya que permiten predecir el comportamiento de la realidad ante el cambio o la supresión de alguna de las variables intervinientes ó ante la introducción de una o de más nuevas variables.

El modelo será tanto más representativo de la realidad, cuanto mayor haya sido el análisis de las variables intervinientes así como de sus relaciones.

Debido a la complejidad de los fenómenos de la naturaleza, casi siempre, al construir un modelo matemático, se tienen que hacer simplificaciones que hacen que esta imagen matemática sea una “aproximación” a la realidad que trata de reproducir. Por lo mismo, se debe distinguir claramente entre el modelo como representación y la realidad misma.

Dos ejemplos de esta situación se plantean en el estudio de la física en educación media cuando en el movimiento de proyectiles y en la caída libre “se descarta la resistencia del aire” y cuando, en el estudio del péndulo se suspende una masa de “un hilo inextensible y sin masa”. De allí que, las deducciones y proyecciones que se hagan con base en estos modelos deban ser tratadas con la debida cautela. Todo modelo es intrínsecamente aproximado y toda conclusión que se base en él debe tener en cuenta esta característica¹.

Clasificación de los modelos: El tipo de modelo utilizado para representar una realidad depende de esa realidad. No toda realidad puede ser representada por un mismo tipo de modelo. De allí que deba destacarse que existen varias clasificaciones para los modelos, de acuerdo con el criterio que se considere. En cuanto a la concepción general del conocimiento se puede hablar de modelos físicos, esquemáticos y simbólicos.

También se puede clasificar los modelos con base en diferentes criterios que se presentan en su representación o construcción, de la siguiente manera²:

- Cambio en atributos y variables dependientes

¹ CHAMORRO VICENTE, REVELO OSCAR, “Modelado: Un Primer Contacto”, En Desarrollo

² *Ibíd.*

- Estáticos: no tienen cambios
 - Dinámicos: existen cambios
- Forma en que ocurren los cambios
 - Continuos: cambios suaves con respecto al tiempo
 - Discretos: cambios notorios con respecto a un tiempo
- Por la presencia de variables aleatorias
 - Determinísticos: todo está definido, los valores predecibles
 - Estocásticos: se definen mediante comportamiento de probabilidad
- Por la forma en que avanza el tiempo
 - Intervalos Fijos: el tiempo avanza siempre en la misma cantidad
 - Secuencia de eventos: el tiempo avanza cada vez que sucede un evento
- Forma de representación
 - Físicos: utilizan elementos físicos en su construcción
 - Simbólico: pueden ser ecuaciones o una representación gráfica o lógica de la situación, como un programa
- Representación del factor humano
 - Suaves: representan el comportamiento de personas
 - Duros: no representan el comportamiento de seres humanos
- Modelo conceptual: es el resultado de haber analizado el sistema y la forma en que el diseñador percibe la situación y decide representar la operación del sistema

Elementos estructurales del modelo:

- Técnicamente, los elementos del modelo se llaman³:
 - Entidades: elementos a ser procesados en el sistema
 - Locaciones: lugares donde se procesan las entidades
 - Recursos: agentes o métodos usados para procesar las entidades

³ *Ibíd.*

- Rutas: trayecto que siguen las entidades una vez que ingresan al sistema a recibir su servicio o ser procesadas
- Entidades: son los objetos procesados en el sistema, representan las entradas y salidas
- Cada una de ellas tiene ciertos atributos, por ejemplo pueden tener un valor que indique si un producto se dictaminó como defectuoso o no defectuoso
- Entidades: De cada entidad se colectan en la simulación, varios valores estadísticos que indican el comportamiento del sistema mediante lo que sucede en el modelo, por ejemplo, tiempo promedio que cada entidad estuvo en espera por el servicio
- La decisión de cuántas y cuáles entidades son importantes en el modelo para representar el sistema depende del propósito por el cual se está haciendo la simulación
- Locaciones: son los lugares que son visitados por las entidades durante su estancia en el sistema ya sea en proceso, espera o en lugares en que se debe hacer una decisión
- Normalmente habrá especial interés en colectar información sobre los contenidos promedio de una locación como una fila, o una bodega
- Las locaciones que deben usarse dependerán del propósito del modelo
- Recursos: son las maneras en que se llevará a cabo el proceso o servicio de cada entidad.
- Pueden ser estáticos (como una máquina copiadora) o dinámicos si se mueven de ubicación en el sistema (un operario de una máquina)
- Pueden representar seres animados (personas) o inanimados (máquinas)
- La diferencia con una entidad, que también es dinámica, es que la entidad acude al sistema a recibir un servicio y el recurso pertenece al sistema y está ahí para dar el servicio a la entidad

Modelación matemática: Como se dijo, un modelo matemático es una representación que se basa en relaciones matemáticas para intentar una interpretación útil de una realidad. El objetivo de la modelación matemática es el de generar una representación matemática útil de una situación real. La modelación matemática dispone de tres tipos de representaciones principales estrechamente interrelacionadas: numérica, gráfica y simbólica, que generalmente deben seguir la siguiente secuencia lógica en su construcción:

Realidad → Representación Numérica → Representación Gráfica (si se la considera de utilidad) → Representación Simbólica

Todas las “fórmulas”, gráficas y tablas que se conocen, que describen de alguna manera un sistema real, son modelos matemáticos de realidades.

Y es aquí donde Dionisio adquiere relevancia: existen cientos de algoritmos y paquetes computacionales que resuelven eficientemente los modelos matemáticos que los humanos logran construir partiendo de la redacción escrita de un problema; pero, la existencia se reduce prácticamente a cero cuando se trata de encontrar un paquete computacional ó al menos un algoritmo efectivo que, partiendo de la redacción escrita de ese problema construya su modelo.

Las instituciones que velan por la calidad de la Educación en Colombia y aquellas agremiaciones de profesionales que en el mismo sentido hacen lo propio, han insistido desde hace varios años en la necesidad de incluir la modelación como un área del conocimiento a la que debe dársele toda la importancia.

Dionisio, como prototipo de modelador de problemas de Simplex, es un primer intento de solución computarizada a las graves falencias que se presentan en el momento en que se necesita construir modelos matemáticos del área a partir de la redacción escrita del problema.

Con un poco más de dedicación y de manera muy fácil puede generalizarse este intento a muchos campos del conocimiento, con sólo establecer las reglas semánticas correspondientes⁴.

2.2.2 Programación lineal: Desde mediados de siglo XX el desarrollo de la programación lineal (PL) ha sido notable. Hoy en día es una herramienta utilizada en grandes, medianas y pequeñas compañías, en distintos países del planeta y en diversos sectores de la sociedad, que ven cómo gracias a esta herramienta pueden optimizar sus procesos haciéndose cada vez más competitivos en un mundo que así lo demanda.

La programación lineal ha acaparado, en buena medida, la atención científica. En este campo sobresalen programas de computadoras dedicados al uso de la programación lineal y muchos artículos y libros publicados sobre esta materia. En teoría, *“La programación lineal es una técnica de investigación de operaciones para la determinación de la asignación óptima de recursos escasos cuando la*

⁴ *Ibíd.*

*función objetivo y las restricciones son lineales.”*⁵ Según esto, se puede decir que, el problema que la programación lineal aborda, se resume en la asignación óptima de recursos limitados a actividades, también conocidas como variables, que compiten por ellos para ser realizadas. Poniéndolo de esta manera, la variedad de situaciones que se adaptan a esta descripción son muchas, desde la planeación presupuestal en una casa de familia hasta la planeación presupuestal en una nación; desde la programación de una dieta de comida hasta la programación de complicadas mezclas químicas, pasando por la planeación de transportes en una pequeña ciudad o la planeación de las rutas aéreas de una aerolínea de talla mundial, sólo por citar unos poquísimos ejemplos.

No obstante y aunque estas áreas son muy distintas entre sí, la programación lineal es transversal a todas ellas debido a que en todas estas situaciones el objetivo es optimizar la utilización de los recursos disponibles para llevar a cabo los diferentes procesos.

Para abordar y solucionar este tipo de problemas, la programación lineal ha desarrollado diferentes metodologías, cada una de las cuales tiene como requisito la construcción de un modelo matemático para describir la situación que se pretende resolver. Hasta ahora la habilidad de construir dichos modelos ha sido exclusiva de los seres humanos y por ello se aspira a que, a partir de Dionisio, se empiece a dotar de esta habilidad a las máquinas.

Modelo de programación lineal: Antes de definir un modelo de programación lineal es conveniente establecer la terminología y notaciones básicas de los elementos que componen estos modelos, según Hillier y Lieberman⁶:

- Variables: En los problemas de programación lineal son las actividades que realiza el sujeto del problema, usualmente una empresa productora, un inversionista o una empresa que vende o compra.
- Recursos: Son los elementos que el sujeto del problema utiliza para llevar a cabo las actividades o variables, como materias primas, dinero o máquinas.

⁵ MOSKOWITZ HERBERT, WRIGHT GORDON, “Investigación de Operaciones”, 1ª Ed., PRENTICE - HALL, México D.F., 1982.

⁶ FREDERICK S. HILLIER, GERALD J. LIEBERMAN, “Introducción a la Investigación de Operaciones”, 8ª Ed., Mc GRAW-HILL, México D.F., 2006.

- Nivel de Variable: Es el resultado que la programación lineal desea encontrar; básicamente el número o cantidad de elementos de una variable, sujeto a unos objetivos y a unos recursos limitados.
- Función Objetivo: Es el valor que se quiere optimizar. Esta optimización está sujeta a los niveles de variable y a la limitación de los recursos.

Los anteriores términos suelen denotarse en los diferentes textos de Investigación de Operaciones de la siguiente manera:

- Z : Valor de la medida global de efectividad o función objetivo.
- x_j : Nivel o cantidad de la actividad j (para $j = 1, 2, \dots, n$).
- c_j : Incremento en Z que se obtiene al aumentar una unidad en el nivel de la actividad j .
- b_i : Cantidad de recurso i disponible para asignar a las actividades (para $i = 1, 2, \dots, m$).
- a_{ij} : Cantidad del recurso i consumido por cada unidad de la actividad j .

El modelo plantea el problema en términos de toma de decisiones sobre los niveles de las variables, por lo que $x_1, x_2, x_3 \dots x_n$ se llaman variables de decisión. Las c_j, b_i y a_{ij} se conocen como constantes o parámetros del modelo.

Ahora se formula el modelo matemático del problema general de asignar recursos a variables, o sea el modelo matemático que Dionisio generará a partir del texto del problema. En particular, este modelo consiste en elegir valores de x_1, x_2, \dots, x_n para:

$$\text{Maximizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \end{aligned}$$

y

$$x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n \geq 0$$

Ahora se puede resumir la estructura de los modelos de programación lineal así:

- **Función objetivo:** Está compuesta por un objetivo, en este caso Maximizar, y por una ecuación que es el valor que se quiere optimizar.

$$\text{Maximizar} \quad Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

- **Restricciones funcionales:** Son la limitaciones de los recursos. Usualmente por cada limitante de los recursos aparece una restricción.

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

- **Restricciones de no negatividad:** Son restricciones inherentes a las variables ya que éstas, para este tipo de problemas, no pueden ser negativas.

$$x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n \geq 0$$

A esta forma general que se acaba de describir, donde el objetivo es maximizar Z ; las restricciones funcionales son desigualdades con sentido menor o igual; las variables de decisión tienen restricciones de no negatividad y b_j no negativos, se la denomina forma estándar del modelo.⁷

Sin embargo existen otras formas de modelos de problemas de programación lineal que Dionisio puede construir y que se describen a continuación:

- Maximizar Z ; restricciones funcionales con desigualdad en sentido mayor o igual; variables de decisión con restricciones de no negatividad y b_j no negativos:

$$\text{Maximizar} \quad Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

⁷ *Ibíd.*

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\geq b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\geq b_2 \\
&\dots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\geq b_m \\
&\text{y} \\
x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n \geq 0
\end{aligned}$$

- Maximizar Z , restricciones funcionales en forma de ecuación, variables de decisión con restricciones de no negatividad y b_j no negativos:

$$\text{Maximizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\
&\dots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \\
&\text{Y} \\
x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n = 0
\end{aligned}$$

- Maximizar Z , algunas restricciones funcionales con desigualdad en sentido mayor o igual, algunas restricciones funcionales con desigualdad en sentido menor o igual, algunas restricciones en forma de ecuación, variables de decisión con restricciones de no negatividad y b_j no negativos:

$$\text{Maximizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\geq b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\
&\dots \\
a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \\
&\text{Y}
\end{aligned}$$

$$x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n \geq 0$$

- Minimizar Z , restricciones funcionales con desigualdad en sentido menor o igual, variables de decisión con restricciones de no negatividad y b_j no negativos:

$$\text{Minimizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \end{aligned}$$

$$\text{Y} \\ x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n \geq 0$$

- Minimizar Z , restricciones funcionales con desigualdad en sentido mayor o igual, variables de decisión con restricciones de no negatividad y b_j no negativos:

$$\text{Minimizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\geq b_2 \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\geq b_m \end{aligned}$$

$$\text{Y} \\ x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n \geq 0$$

- Minimizar Z , restricciones funcionales en forma de ecuación, variables de decisión con restricciones de no negatividad y b_j no negativos:

$$\text{Minimizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &= b_m \end{aligned}$$

Y

$$x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n = 0$$

- Minimizar Z , algunas restricciones funcionales con desigualdad en sentido mayor o igual, algunas restricciones funcionales con desigualdad en sentido menor o igual, algunas restricciones en forma de ecuación, variables de decisión con restricciones de no negatividad y b_j no negativos:

$$\text{Minimizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Sujeta a las restricciones:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\ &\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \end{aligned}$$

Y

$$x_1 \geq 0, \quad x_2 \geq 0, \quad \dots \quad x_n \geq 0$$

Estos son los tipos modelos que Dionisio construirá a partir de texto que el usuario digite. Se hace necesario mencionar que, además de estas formas de modelo, existen formas especiales donde algunas o todas las variables de decisión no tienen restricciones de no negatividad o incluso donde aparecen valores negativos para algunos o todos los b_j . Este tipo de modelos no serán abordados por Dionisio.

Método simplex: Después de plantear el modelo se debe resolver el sistema matemático resultante. Para solucionarlo se han desarrollado diversos métodos, entre ellos el Método Simplex, procedimiento desarrollado por George Dantzig en 1947, para resolver problemas de programación lineal.

Dicho procedimiento busca resolver cualquier modelo de programación lineal que se establezca en la forma estándar descrita anteriormente. En caso de que un modelo no se ajuste a esta forma, existen técnicas para ajustarlos a ella, como la variación Dual del Método Simplex y la Técnica de la Dos Fases.

Básicamente el Método Simplex es un algoritmo iterativo que organiza los datos del modelo de programación lineal en una matriz que se soluciona a través del método de Gauss – Jordan, encontrando en cada paso una posible solución que puede ser óptima si cumple con ciertas condiciones del proceso. De no serlo, se repite el proceso hasta encontrar dicha solución.

En este trabajo, el desarrollo del algoritmo del Método Simplex al igual que el de sus variaciones o técnicas auxiliares no es el centro del estudio. Por esta razón no se profundiza más en su teoría y desarrollo. De ahora en adelante el término Simplex será utilizado para clasificar el tipo de modelos que Dionisio construirá.

2.2.3 Inteligencia artificial: Sobre la década de 1950 y, en gran medida gracias al desarrollo de las computadoras, aparece una disciplina que tiene como objeto de estudio las entidades inteligentes con el fin de comprender cómo funcionan, generando un conocimiento a partir del cual se puedan crear nuevas entidades igual o más inteligentes a las estudiadas. Nació así la Inteligencia Artificial (IA).

Rápidamente esta nueva área del conocimiento acaparó la atención de muchos estudiosos de diferentes campos de la ciencia, debido a que resultaron de gran interés los diversos y novedosos productos que la IA proponía. Además, al ser nueva, ofrecía y ofrece un gran campo de acción donde los diferentes científicos pueden ser reconocidos por sus grandes aportes, en contraposición a la de antiguas ciencias como la física, donde proponer una idea revolucionaria implica años de estudio, con un alto riesgo de conseguir nada, debido a que las grandes ideas de la física “ya han sido propuestas”.

Conviene ahora definir los objetivos de la IA. Al construir un agente inteligente, básicamente estos objetivos se encasillan en cuatro enfoques adoptados

históricamente por diferentes autores, tal es el caso de Russel y Norvig⁸ que definen estos enfoques más o menos así:

- **Actuar como humano:** Este enfoque dice que una entidad inteligente creada por un ser humano como lo es una computadora debe ser capaz de imitar a un humano en todas las actividades que éste realiza desde su percepción del mundo hasta la forma de efectuar acciones sobre él, pasando por cómo infiere el quehacer ante las circunstancias percibidas. Para esto se plantea una prueba que consiste en que la máquina debe interactuar con un humano, el cual debe creer en el proceso, que esta interactuando con otro humano igual a él. A esta prueba se la conoce como prueba de Turing y para ser aprobada por la máquina ésta deberá ser capaz de procesar lenguaje natural ya sea hablado o escrito, representar el conocimiento, razonar automáticamente y aprender por sí sola. Todo esto, en caso de que sólo se quiera comprobar inteligencia. Si, además se quieren comprobar también características físicas, la máquina deberá estar dotada de sistemas sensitivos que le permitan percibir y de efectores que le permitan actuar.
- **Pensar como humano:** Este enfoque se basa principalmente en que un agente inteligente artificial imite la forma de razonar de un humano. Para esto se necesita previamente conocer el modo en que éste razona, lo cual se puede lograr por medio de dos técnicas: una, llamada introspección, que consiste en adentrarse en la mente y atrapar los pensamientos conforme éstos se vayan produciendo, y otra, por medio de la realización de experimentos psicológicos.

Después de recopilar esta información sobre el proceso del pensamiento humano, se procede a plasmar dicho conocimiento en un programa de computadora. Para comprobar que la computadora imita la forma de pensar de un humano, los datos de entrada y salida del programa deben ser los mismos que utiliza y produce el humano. Además los tiempos de procesamiento de la información y obtención de resultados de las dos entidades también deben ser similares.

- **Pensar racionalmente:** Partiendo de la definición aristotélica de pensamiento racional que afirma que pensar racionalmente es “la manera correcta de pensar”, este enfoque se basa en la creación de agentes inteligentes que piensen lógicamente (que piensen de manera correcta), donde el agente obtenga las respuestas correctas ante una circunstancia, siempre y cuando se

⁸ STUART RUSSELL, PETER NORVIG, “Inteligencia Artificial un Enfoque Moderno”, 1ª Ed, PRENTICE HALL, México D.F., 1996

parta de las premisas correctas. Para ello el agente debe tener la capacidad de traducir a notación lógica el problema planteado y encontrarle solución si ésta existe.

Debido a lo idealista que resulta el concepto de pensamiento racional este enfoque presenta grandes inconvenientes a la hora de ponerlo en práctica. Estos consisten básicamente en la dificultad de expresar formalmente el conocimiento que se presenta el mundo real ya que éste tiene un nivel de incertidumbre considerablemente alto. Por otro lado, resolver un problema lógicamente puede resultar muy fácil en teoría, pero al momento de pasar a la práctica, son muchos los inconvenientes que una computadora presentaría, ocasionados por su relativa baja capacidad de procesamiento.

- **Actuar en forma racional:** Según este enfoque, la IA tiene como objetivo el estudio y construcción de agentes racionales o agentes que actúen racionalmente; esto quiere decir, agentes que alcancen las metas propuestas con base en cierto conocimiento supuesto sin tener en cuenta la forma cómo se infirió la manera de actuar. Ésta es la principal diferencia con el anterior enfoque que, ante todo, busca la forma correcta de inferir; ahora según este enfoque, en muchas ocasiones pueden haber formas que no sean correctas pero que deban llevarse a cabo para lograr los objetivos deseados por lo que la forma de actuar no siempre depende de la lógica. Además expone situaciones como las de actos reflejos donde la forma racional de actuar no requiere de ninguna inferencia.

A partir de estos enfoques propuestos por diferentes autores en escritos acerca de IA se puede decir que el principal objetivo de esta novedosa disciplina es la creación de agentes inteligentes por parte del ser humano. Dionisio pretende ser un agente inteligente que simula la forma en que los humanos modelan problemas de Programación Lineal. Por ello, Dionisio encaja en el enfoque de pensar como humano y esto se sustenta en que las entradas y salidas del proceso de modelado en Dionisio son las mismas que las del proceso de modelado en un humano. Por otra parte se encuentra que Dionisio puede modelar un problema en menos tiempo que un ser humano, basándose en el método de introspección anteriormente mencionado, por medio del cual se examina la forma en que la mente humana modela un problema de programación lineal. A partir de esto se genera una serie de reglas y acciones que Dionisio realiza para lograr su objetivo; estas reglas y acciones serán explicadas más adelante en este mismo documento.

El desafío de la IA es tan interesante como difícil. Por ello dentro de la misma, se han creado muchas subdisciplinas que, a pesar de su corta existencia, podrían ser

consideradas como independientes. A continuación se profundiza en la definición y caracterización de los subcampos de la IA que se aprovechan para construir Dionisio. Son estos:

Sistemas expertos: Un sistema experto es un agente inteligente que simula la forma de pensar de un humano. Esta es la pretensión de Dionisio y se requiere por tanto, profundizar en la definición y principales características y componentes del Sistema Experto.

El área de estudio de los Sistemas Expertos (SE), se origina en las técnicas empleadas por la IA desde sus principios. A pesar de esto, sólo a mediados de la década de los sesenta se empieza a realizar una investigación específica en este campo. En primera instancia los aplicativos creados resultaron muy limitados como consecuencia de la escasa tecnología con la que se contaba en la época, además de que estos aplicativos estaban orientados a temas de alto carácter científico por lo que no tuvieron gran reconocimiento. Sin embargo, con el pasar del tiempo y el avance de la tecnología, los SE han sido cada día más útiles en las diferentes actividades humanas y se espera que con el tiempo se sigan posicionando como una herramienta más en la vida cotidiana.

Según David W. Rolston: *“Un Sistema Experto es una aplicación informática que soluciona problemas muy complicados que de otra manera exigirían ampliamente la pericia humana. Para lograr esto, se simula el proceso de razonamiento humano mediante la aplicación específica de conocimientos y de inferencias.”*⁹

A partir del concepto de Rolston, se puede decir que el principal objetivo de los SE es crear agentes capaces de realizar tareas que normalmente son ejecutadas por personas expertas bien entrenadas a través de la capacitación y los años de experiencia. Para lograrlo, el principio de los SE es captar el conocimiento que le permite a la persona actuar de cierta manera ante determinadas circunstancias. Esto hace que este tipo de sistemas tengan un atractivo particular sobre los demás aplicativos informáticos ya que, por medio de ellos, las máquinas reflejan muchas características de la forma de pensar humana.

⁹ DAVID W. Rolston, “Principios de Inteligencia Artificial y Sistemas Expertos”, 1ª Ed., Mc GRAW-HILL, México D.F., 1992.

Igualmente, Rolston establece que, Para alcanzar el objetivo de simular la forma de pensar de un humano, un SE debe estar conformado por los siguientes elementos¹⁰:

- **Interfaz de Usuario:** como interfaz de procesos de entrada, este elemento debe ser capaz de tomar la información del usuario y convertirla en datos con los que el resto del sistema pueda trabajar. En el proceso de salida, este elemento debe ser capaz de presentar los datos generados por el SE, como información entendible para el usuario.
Este proceso usualmente se lleva a cabo con sistemas de procesamiento de lenguaje natural. Cabe mencionar que hasta ahora no existe un sistema con la habilidad de reproducir las capacidades del lenguaje en su totalidad ya que ha resultado un reto difícil de sortear.
- **Base de Conocimientos:** Como su nombre lo indica este módulo es el que almacena el conocimiento disponible para el sistema, básicamente en forma de hechos y reglas.
- **Motor de Inferencia:** Este módulo permite a los SE inferir nuevo conocimiento a partir del existente.
- **Medios para Actualización de Conocimientos:** A través de estos medios el usuario puede actualizar los conocimientos con los que trabaja el SE, básicamente por inserción de nuevos hechos y reglas o por modificación de los existentes.
- **Medios para la Explicación:** A través de estos medios el SE explica que línea de razonamiento siguió para llegar a la respuesta obtenida.

Como se mencionó anteriormente y como se detallara más adelante, Dionisio debe contar con muchas de las características que definen y conforman a un SE. De cierta forma Dionisio aprovecha algunos de los elementos listados arriba para llevar a cabo su proceso de modelado. Por ejemplo, cuenta con un sistema de procesamiento de lenguaje natural que lleva a cabo los procesos más importantes dentro del sistema modelador y, por otro lado, Dionisio parte de hechos y reglas, determinados a través de la introspección ya citada, para llevar a cabo el

¹⁰ *Ibíd.*

procesamiento del lenguaje, la extracción de elementos del texto y la construcción del modelo.

En cuanto a los módulos restantes que conforman un SE aunque podrían ayudar a robustecer en mucho a Dionisio, no se han implementado aún, ya que el objetivo fundamental fue hacer que Dionisio modele ejercicios de programación lineal y no fueron necesarios para esta etapa. Sin embargo, la creación del motor de inferencia como la de los medios para la actualización de conocimientos, son los primeros trabajos que esta investigación plantea hacia el futuro.

Procesamiento de lenguaje natural: El Procesamiento de Lenguaje Natural (PLN), es sin duda uno de los mayores desafíos que aborda la IA en la búsqueda de crear agentes inteligentes que puedan interactuar con seres humanos, imitando sus habilidades, debido a que pretende dotar a las máquinas de comprensión inteligente del lenguaje, con el fin de facilitar la interacción entre ellas y el hombre.

A mediados del siglo XX, cuando aparecieron las primeras aplicaciones de PLN, el interés fundamental era lograr la traducción automática. La primera propuesta fue la sustitución de una palabra por su equivalente en el otro idioma pero esto dio resultados muy limitados, por lo que fue necesario tener en cuenta la sintaxis y la semántica del idioma así como la información contextual inmersa en los textos. Así, un problema que al principio parecía tener una solución simple se complicó, ya que las ambigüedades sintácticas y semánticas, además de la falta de orden en la escritura son dificultades que hasta hoy no se han podido resolver plenamente.

Para lograr una solución a los problemas planteados, el PLN incorporó a su estudio diversas áreas del conocimiento. Algunas décadas después se puede encontrar entre estas áreas la lingüística, la ingeniería informática, la filosofía, las matemáticas y la psicología, las que unidas, emplean técnicas de IA para cumplir con sus objetivos.

En este proceso se ha establecido que el estudio del procesamiento del lenguaje natural debe abordarse desde cuatro niveles de análisis¹¹: el morfológico, el sintáctico, el semántico y el pragmático, que se describen a continuación:

¹¹ [En línea] EL PROFESIONAL DE LA INFORMACION – Revista Internacional Científica y Profesional – ESPAÑA.

<http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actual_bases_tericas_y_aplicaciones_parte_i.html> [Consulta:23 de Junio de 2010]

- **Análisis morfológico:** Por medio de este análisis se busca determinar las relaciones que se establecen entre las unidades mínimas que conforman una palabra como lo son la raíz de la palabra, los sufijos y prefijos. Para ello se relaciona directamente con el léxico, que es el conjunto de información sobre cada palabra que el sistema utiliza.
Dionisio realiza un análisis morfológico del texto en el momento de comparar palabras ya que para lograrlo, se utilizan las raíces de las palabras. Por otra parte, se estableció desde el principio del proyecto que el léxico empleado es el de Investigación de Operaciones y más específicamente, el de programación lineal en problemas de Simplex.
- **Análisis sintáctico:** Este análisis tiene como objetivo determinar la combinación correcta de las palabras. Para ello, etiqueta cada elemento del texto y a partir de estos elementos etiquetados, genera una estructura que corresponda con cada posible categoría sintáctica.
- **Análisis semántico:** Su objetivo es determinar el significado del texto que se quiere tratar, tarea que no resulta nada sencilla ya que aquí es donde se debe afrontar la ambigüedad del lenguaje. Esta ambigüedad hace que un mismo texto pueda dar lugar a diversas interpretaciones. En este caso, la semántica intenta proporcionar el significado del texto independientemente del contexto, o sea que se trabaja con el significado individual de cada palabra. Esto hace que los resultados obtenidos en este análisis hasta hoy, sean muy pobres, ya que no tiene en cuenta la valiosa información del contexto.
- **Análisis pragmático:** Al igual que el análisis semántico, este análisis pretende determinar el significado del texto que está tratando. La diferencia radica en que el Análisis Pragmático sí tiene en cuenta el contexto en el que la palabra aparece, es decir la frase a la que pertenece y para ello contempla las relaciones existentes entre las palabras de la frase, además de tener en cuenta la evolución del discurso, que se ocupa en el cómo frases anteriores del texto, pueden influir en el significado de la que se está tratando.
Dionisio aplica el Análisis Pragmático ya que una palabra o lexema puede variar su significado dependiendo de la clasificación del subpárrafo al que pertenezca. Para lograrlo se utilizaron conceptos de lógica difusa que ayudan a categorizar los lexemas que presentan ambigüedad.

Resumiendo, Dionisio utiliza conceptos del análisis morfológico y pragmático del lenguaje natural, así como conceptos de la evolución del discurso (herencia del significado), para lograr abstraer la mayor cantidad posible de datos del texto redactado por el usuario.

Lógica difusa: También conocida como lógica borrosa, esta disciplina matemática ha despertado en los últimos años un gran interés, ya que permite tratar con la incertidumbre y lo impreciso de la mayoría de los objetos del mundo real, lo que ha hecho posible su aplicación en diversos campos que van desde complicados autómatas hasta aparatos de uso cotidiano.

El concepto de lógica difusa aparece por vez primera a mediados de la década de 1960 cuando en la Universidad de Berkeley el ingeniero Lofty A. Zadeh, postuló que el pensamiento humano no se construye sólo a través de datos cuantitativos o numéricos sino que también se utilizan datos cualitativos que, al no ser tan precisos como los números, permiten al razonamiento humano contar con información extra. Estos datos cualitativos son conocidos como etiquetas lingüísticas, las cuales le dan a la lógica difusa la capacidad de producir resultados exactos a partir de datos imprecisos por lo que es de gran utilidad para programas de computadora que manejan este tipo de datos, como en el caso de Dionisio, que tiene que tratar con lo impreciso y la incertidumbre del lenguaje natural.

La lógica difusa se basa en los siguientes conceptos:

- Conjuntos difusos: De acuerdo con Martin del Brío: *"...la teoría de conjuntos difusos parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia al conjunto, definida ésta como un número real entre 0 y 1."*¹² El ejemplo típico con el cual se describe este concepto es el de la estatura, en el conjunto clásico una persona puede pertenecer al conjunto de los altos o al de los bajos calificándolos con valores de 1 y 0 respectivamente, mientras que si se trabaja con lógica difusa, se pueden establecer nuevas categorías intermedias utilizando los valores entre 0 y 1. Por ejemplo, una persona puede ser de estatura media tirando a alta si su grado de pertenencia a media es de 0,6 y su grado de pertenencia a alta es de 0,4. Además, los conjuntos difusos admiten operaciones que han sido catalogadas como operaciones difusas. Entre ellas se encuentran, complemento, unión e

¹² DEL BRIO BONIFACIO MARTIN, SANZ MOLINA ALFREDO, "Redes Neuronales y Sistemas Difusos", 2ª Ed., ALFAOMEGA GRUPO EDITOR– RA - MA, Madrid – España, 2002

intersección, a partir de las cuales se puede dar tratamiento a estos conjuntos en un aspecto matemático¹³.

- Función de inclusión o pertenencia: esta función es la que asigna los valores o grados de pertenencia a cada elemento del conjunto según corresponda. En lógica difusa existen unas funciones estándar para llevar a cabo dicha asignación ya que son las que con mayor frecuencia se presentan. Entre estas se encuentran, las funciones de tipo trapezoidal, singleton, triangular, S, exponencial y tipo π . Cabe anotar que el hecho de que existan estos estándares no impide que se creen nuevas funciones que pueden ser válidas para definir a un conjunto difuso.
- Variable o etiqueta lingüística: Son variables que pueden tomar por valor términos del lenguaje natural y lo que hacen es etiquetar intervalos de pertenencia con una palabra de lenguaje natural. Por ejemplo, en el caso de las estaturas, el intervalo entre 0,8 y 1 recibe la etiqueta o variable lingüística de “alto”. Se resalta que, en el lenguaje natural, también se manejan números, haciendo que una variable o etiqueta lingüística puede finalmente corresponder a un dato numérico.
- Inferencia borrosa: Como en la lógica clásica esta inferencia se ocupa del razonamiento formal a través de proposiciones. La diferencia radica en que ahora las salidas pueden tomar valores entre falso y verdadero.
- Reglas borrosas: Estas reglas combinan un conjunto borroso de entradas llamado premisas con un conjunto borroso de salida llamado consecuentes a través de cláusulas de tipo si entonces, siguiendo las reglas de la lógica clásica. La única diferencia en este tipo de lógica, como se ha venido insistiendo, radica en los tipos de datos que maneja y genera.

Dionisio utiliza conceptos de lógica difusa cuando clasifica las frases que conforman el texto a partir de un conjunto difuso de patrones, los cuales tienen un grado de pertenencia a una u otra categoría, de acuerdo con una escala asignada por lo diseñadores. A partir de allí, Dionisio acumula estos valores de pertenencia a un subpárrafo, según lo que en éste encuentre, para posteriormente, aplicando una comparación, escoger la categoría con el grado de pertenencia mayor para clasificar el texto.

¹³ *Ibíd.*

3. METODOLOGIA

3.1 Metodología Incremental

El modelo de desarrollo de software incremental es un modelo de software evolutivo que combina la iteración presente en la construcción de prototipos y elementos del modelo lineal secuencial. Esta metodología de trabajo consiste en aplicar secuencias lineales mientras progresa el tiempo; al final de cada secuencia se entrega un incremento que sirve como base para la próxima secuencia.

Cada secuencia se divide en cuatro fases: Análisis, Diseño, Código y Prueba, al final de la cual se entrega un producto completamente operacional.

- Análisis: En esta fase se especifican los requisitos u objetivos que se pretenden satisfacer a través del intento.
- Diseño: En esta fase se planifican los procesos o sistemas con suficientes detalles para permitir su interpretación y realización física, basándose en los objetivos previamente dispuestos.
- Código: Es la realización física del software a través de la planificación realizada.
- Prueba: En esta fase se prueba la realización física del software cotejando sus resultados con los objetivos iniciales.

El desarrollo de Dionisio se realiza utilizando una metodología similar a la utilizada por el modelo incremental. Cada incremento ha sido denominado TesisIntento acompañado de un número que indica la iteración; en cada uno de ellos se lleva a cabo una fase de análisis donde se especifica los objetivos del intento; a continuación se realiza el diseño de lo que se va a hacer para su posterior codificación. Las pruebas se realizan a lo largo de la codificación para observar paso a paso que tan cerca se está de cumplir los objetivos planteados. El cambio de iteración o intento se puede dar básicamente por dos razones: la primera, que todo lo planteado inicialmente se logre y se necesite pasar a otra fase del proyecto; la segunda, que en el intento se estanque el proyecto por lo que se necesita un replanteamiento general y un cambio de fondo en el análisis y diseño. Cada TesisIntento tiene como resultado una nueva versión parcial y funcional de Dionisio que es la base para la siguiente iteración. Para el desarrollo de esta

investigación y la obtención de los objetivos principales de Dionisio fueron necesarias seis iteraciones que van a ser descritas a continuación.

4. DESARROLLO DEL PROYECTO

4.1 TESISINTENTO1

Después de la lectura y documentación obligadas, acerca de las diferentes áreas del conocimiento que se creyeron de utilidad para el desarrollo del proyecto y, después de seleccionar algunas de las diferentes ideas que surgieron a partir de la documentación, se decidió llevarlas a la práctica y surgió el primer intento que se describe a continuación.

4.1.1 Análisis

Objetivos

Para este intento se propusieron los siguientes objetivos:

- Dividir el texto de un problema de SIMPLEX en párrafos, identificando donde empieza cada párrafo.
- Dividir los párrafos resultantes de la división anterior en subpárrafos.
- Después de la división realizar un primer intento de clasificación de subpárrafos, según su contenido.

Con el fin de alcanzar los objetivos propuestos, se propuso construir los módulos descritos, en los siguientes diagramas:

Diagrama de casos de uso

Diagrama general: Como se muestra en la Figura 1 – Diagrama de Casos de Uso General TesisIntento6, en este primer intento, Dionisio, tiene como objetivo principal clasificar el texto que el usuario le brinde como entrada.

Figura 1 - Diagrama de Casos de Uso General TesisIntento1

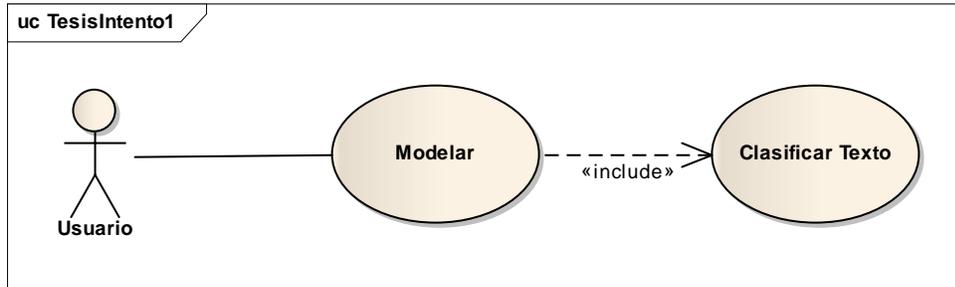


Diagrama específico: Este proceso de clasificación, lleva a cabo las acciones que se detallan en el siguiente diagrama, (Figura 2).

Figura 2 - Diagrama de Casos de Uso TesisIntento1

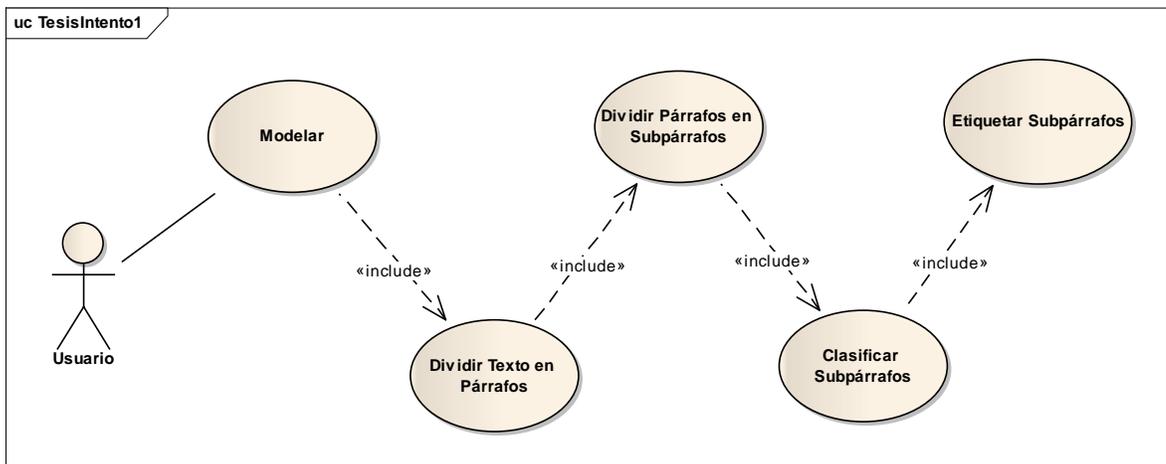
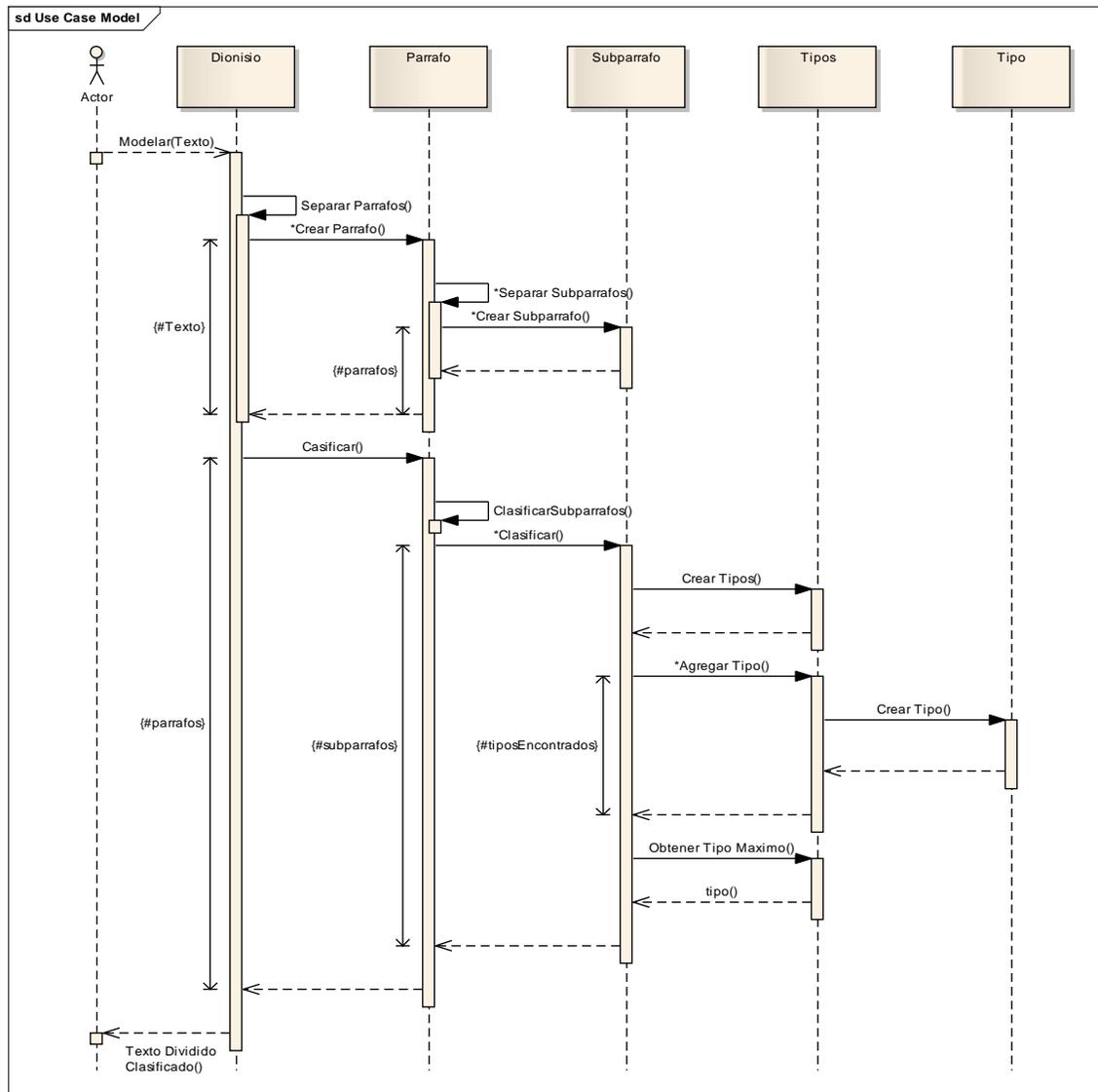


Diagrama de secuencia

Figura 3 - Diagrama de Secuencia TesisIntento1



4.1.2 Diseño

Diagrama general de datos:

A continuación se presenta el diagrama general de datos utilizado en el primer intento (Figura 4).

Figura 4 - Diagrama General de Datos TesisIntento1



Diagrama conceptual: La Figura 5, expresa el modelo conceptual de la base de datos, dilucidando las relaciones entre las tablas y sus principales componentes

Figura 5 - Diagrama Conceptual TesisIntento1

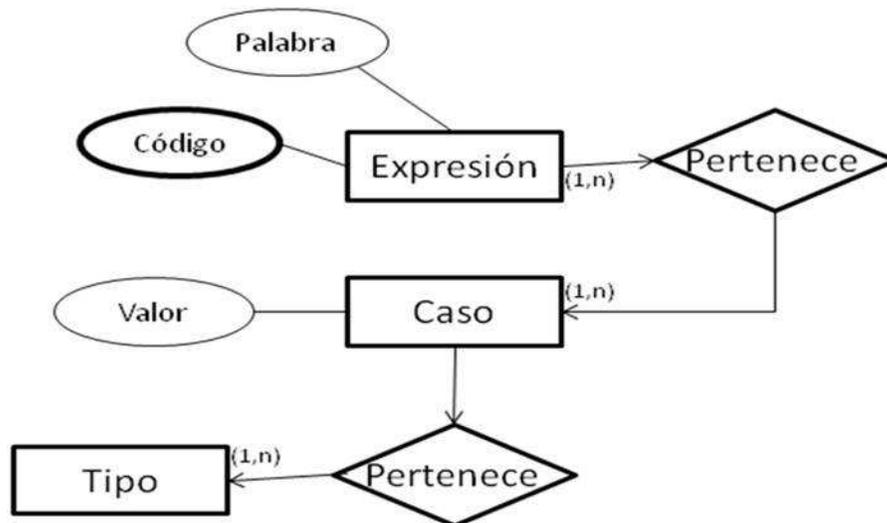


Diagrama de BD: Para lograr el objetivo de clasificar texto, la información es almacenada en la base de datos (Figura 6):

Figura 6 - Diagrama de Base de Datos TesisIntento1

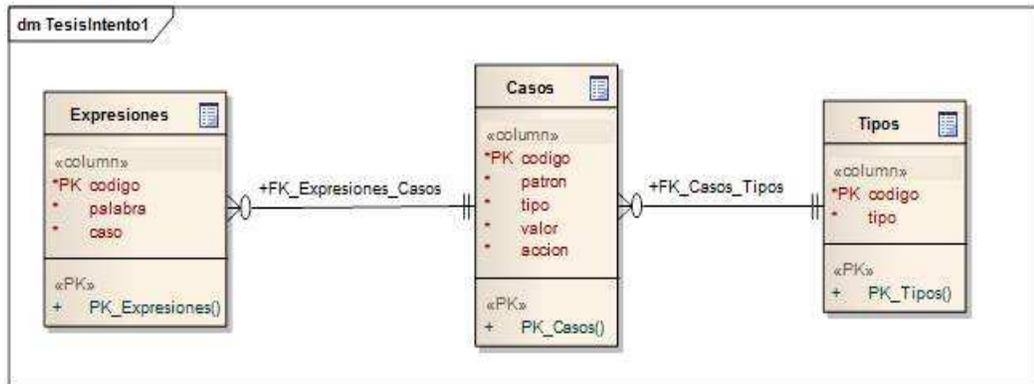


Diagrama de clases: En la Figura 7 – Diagrama de Clases TesisIntento1, se puede observar el diseño inicial planteado para la construcción de Dionisio en su primera iteración.

Figura 7 - Diagrama de Clases TesisIntento1

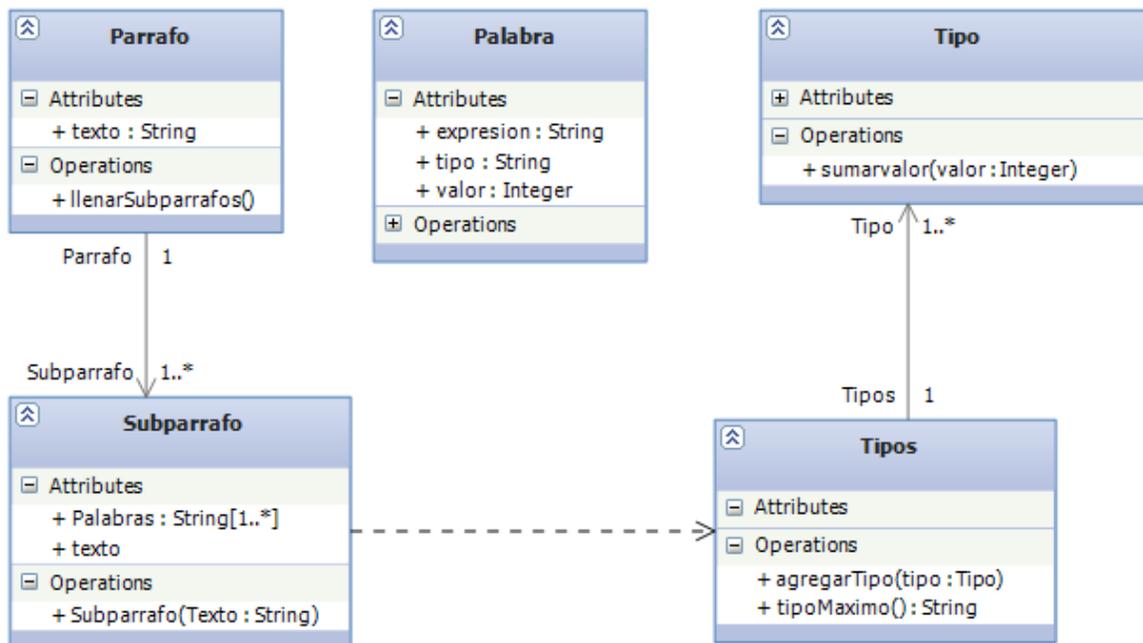
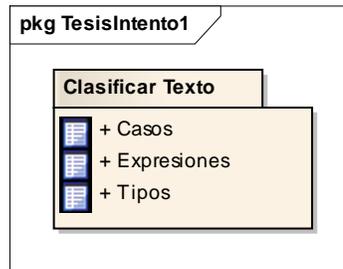


Diagrama de paquetes: Como se puede ver en la Figura 8, las tablas componentes de la base de datos inicial, se agrupan de acuerdo a los módulos o funciones en las que son utilizadas por Dionisio, en este intento la única función que utiliza base de datos es la de clasificación.

Figura 8 - Diagrama de Paquetes TesisIntento1



4.1.3 Resultados obtenidos: El resultado de este primer intento fue una primera versión de Dionisio que realiza una división de texto aceptable y una elemental clasificación. El resultado más importante fue el conjunto de ideas originadas a partir de esta iteración.

Conclusiones

- Este primer intento condujo por primera vez al verdadero enfrentamiento con el problema que se quería solucionar y aportó las primeras impresiones sobre cómo se debía encaminar la investigación, haciendo caer en la cuenta de la real dificultad del problema.
- En la práctica desde este primer intento se definió que la forma en que se debía tratar el texto era dividiéndolo y clasificándolo.

Trabajo posterior: Después de este primer intento, donde se centró la atención en la división y la clasificación del texto, se dedujo que era necesario armar el léxico con el que Dionisio debía trabajar, además de que, para el siguiente intento se debía empezar a extraer información del texto.

4.2 TESISINTENTO2

En este intento se empieza a extraer información del texto del problema. Para ello se necesita mejorar la forma en que Dionisio clasifica las diferentes partes del texto y se mejora la estructura de la base de datos con la que Dionisio debe trabajar.

4.2.1 Análisis

Objetivos

- Establecer el léxico de Investigación de Operaciones con el que trabaja Dionisio.
- Establecer una jerarquía que clasifique las diferentes palabras del léxico de Investigación de Operaciones, de acuerdo con los diferentes componentes de un modelo matemático de programación lineal.
- Codificar algoritmos que permitan a Dionisio obtener el objetivo del problema, buscando variables, recursos y restricciones.

Con el fin de alcanzar los objetivos propuestos, se plantean los siguientes diagramas:

Diagrama de casos de uso

Diagrama general: A continuación (Figura 9), se presentan las acciones que se implementan en este intento con el fin de alcanzar los objetivos propuestos, en comparación con el anterior intento se puede ver que se suman las funciones de obtención de elementos del modelo:

Figura 9 - Diagrama de Casos de Uso General TesisIntento2

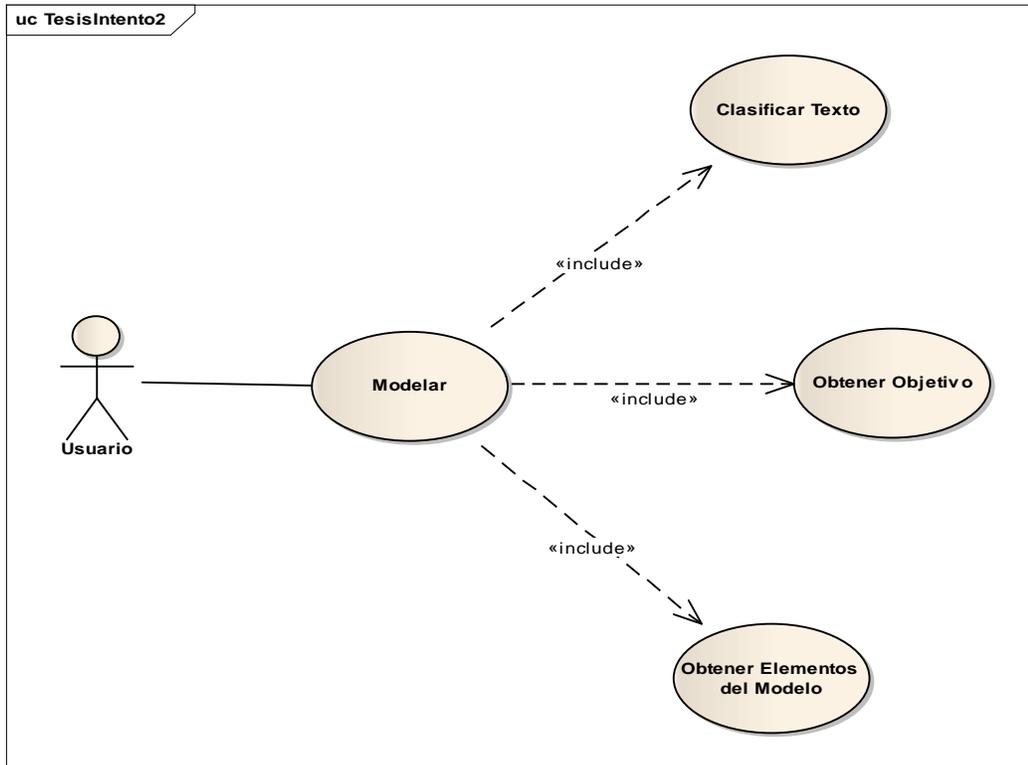


Diagrama específico: Estas nuevas acciones de obtención de elementos y la antigua de clasificación cuentan a su vez con los subprocessos detallados en el siguiente diagrama (Figura 10):

Figura 10 - Diagrama de Casos de Uso TesisIntento2

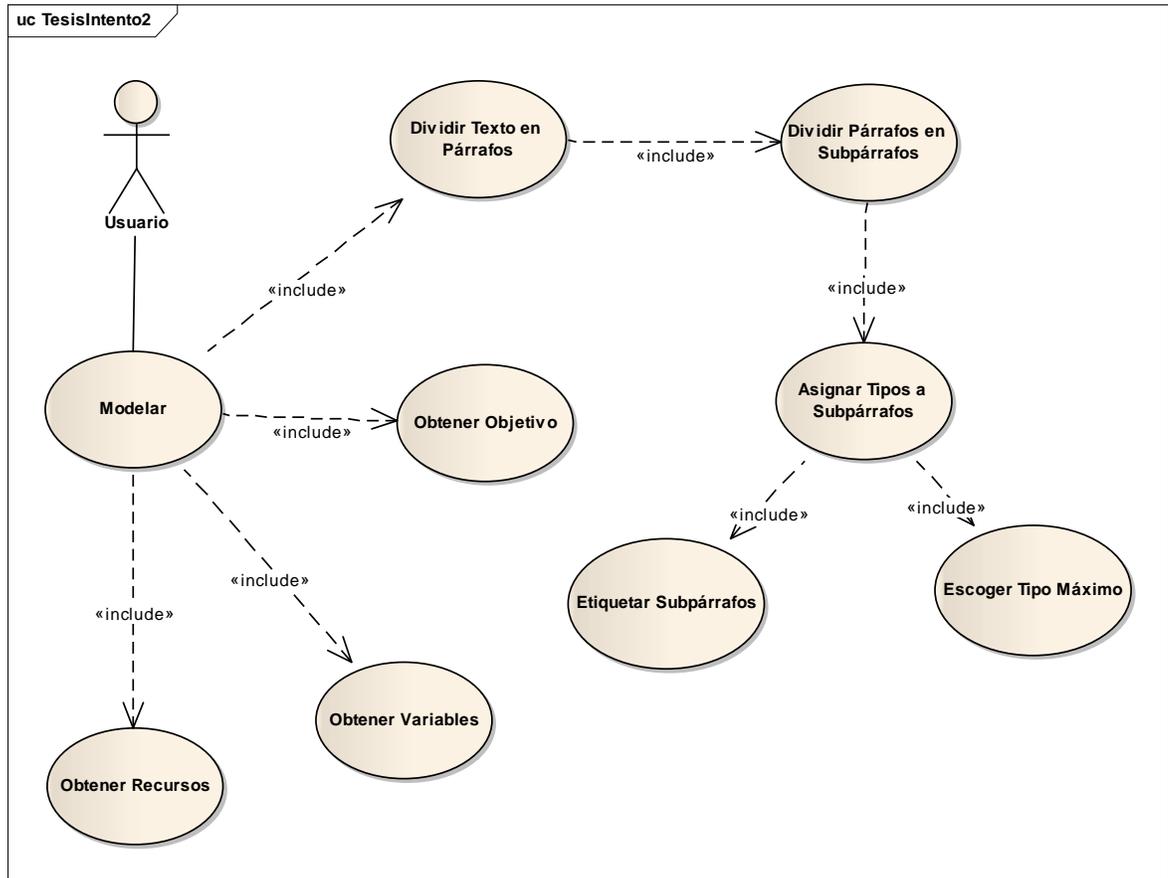
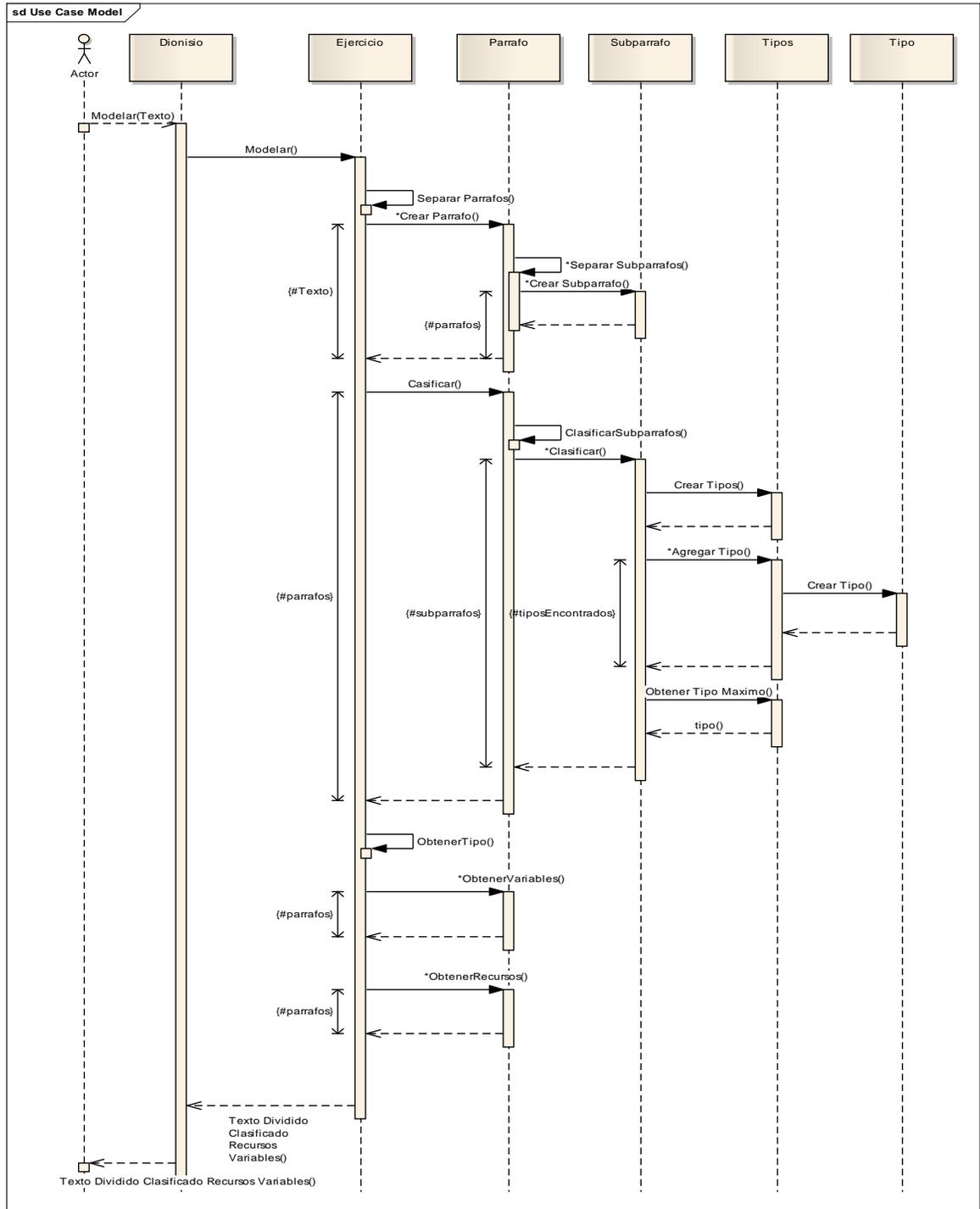


Diagrama de secuencia: En este intento se puede apreciar, el momento en el cual es obtenido el tipo u objetivo del ejercicio, justo entre clasificar subpárrafos y obtener variables, ver Figura 11.

Figura 11 - Diagrama de Secuencia TesisIntento2



4.2.2 Diseño

Diagrama general de datos: Para el segundo intento, como se muestra en el siguiente diagrama (Figura 12), fue necesario agregar nuevas tablas, donde básicamente se almacenaron expresiones propias del léxico de Investigación de Operaciones, esto con el fin de mejorar la clasificación del texto y empezar a dotar a Dionisio de las herramientas necesarias para identificar los elementos del modelo.

Figura 12 - Diagrama General de Datos TesisIntento2

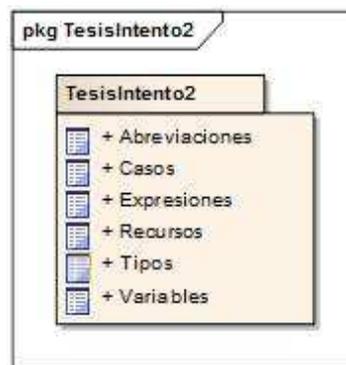


Diagrama conceptual: El cambio principal en este intento es que, la probabilidad que una expresión le da a un subpárrafo de pertenecer a cierto tipo se ubica directamente en la expresión, como se muestra en este diagrama (Figura 13), a través de valor.

Figura 13 - Diagrama Conceptual TesisIntento2

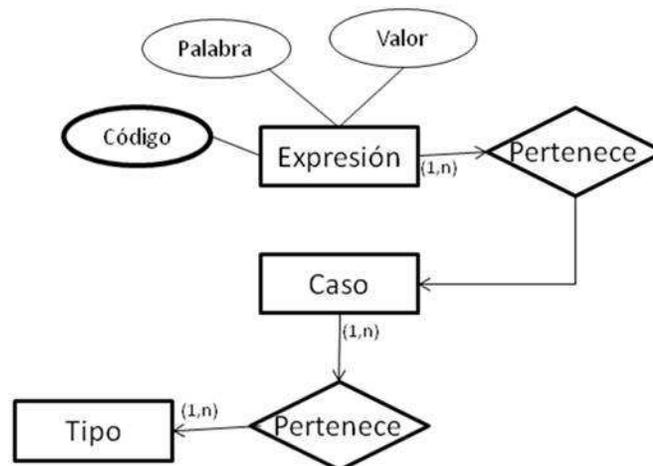


Diagrama de BD: Estas nuevas tablas, se sumaron a las construidas en el primer intento, conformando la estructura que a continuación se presenta (Figura 14), esta estructura es la que implementa la jerarquía de clasificación que se empieza a utilizar desde este intento:

Figura 14 - Diagrama de Base de Datos TesisIntento2

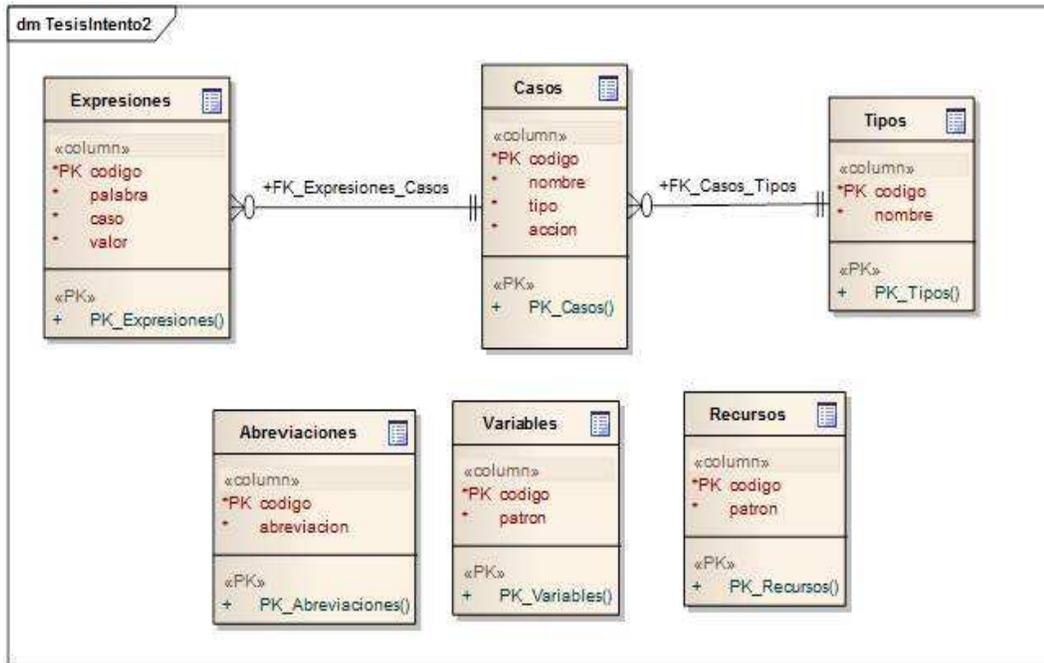


Diagrama de clases: En este intento ya se piensa en la posibilidad de obtener las restricciones de un ejercicio, por eso cuenta con una clase Restricción, ver Figura 15.

Figura 15 - Diagrama de Clases TesisIntento2

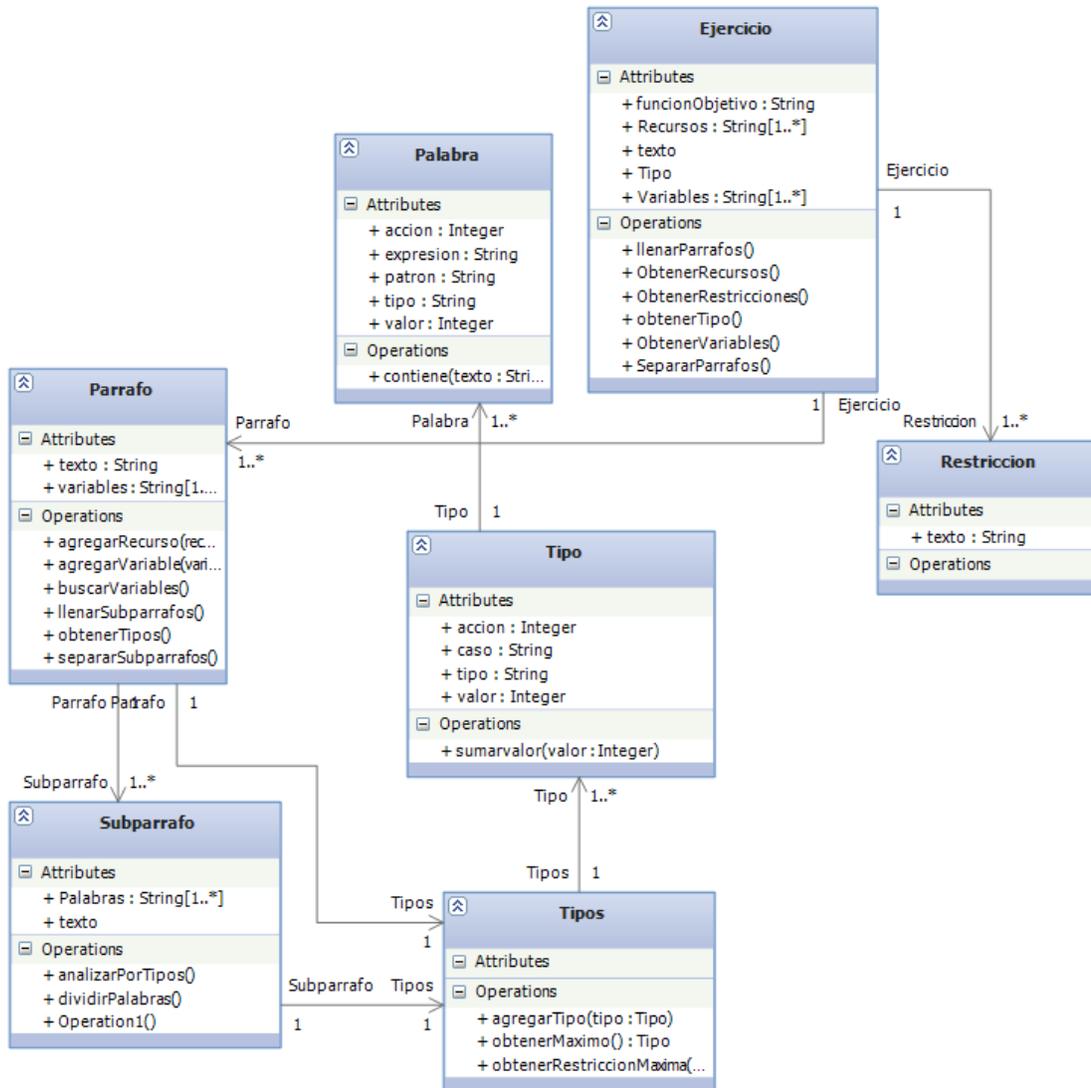
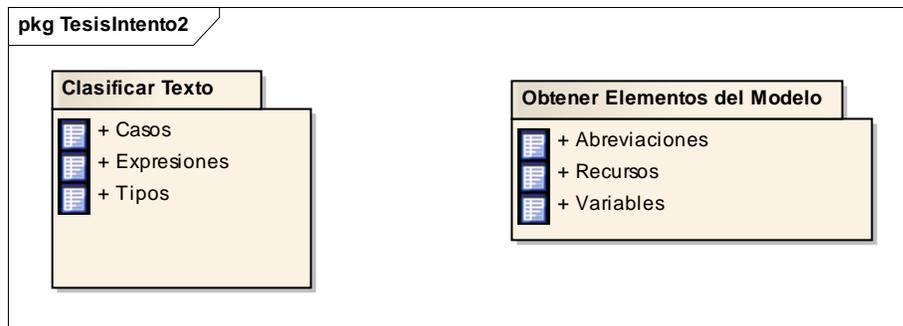


Diagrama de paquetes: En este intento, además de la clasificación de texto, aparece el módulo de obtención de elementos del modelo, estos módulos utilizan la información almacenada de la siguiente manera (Figura 16):

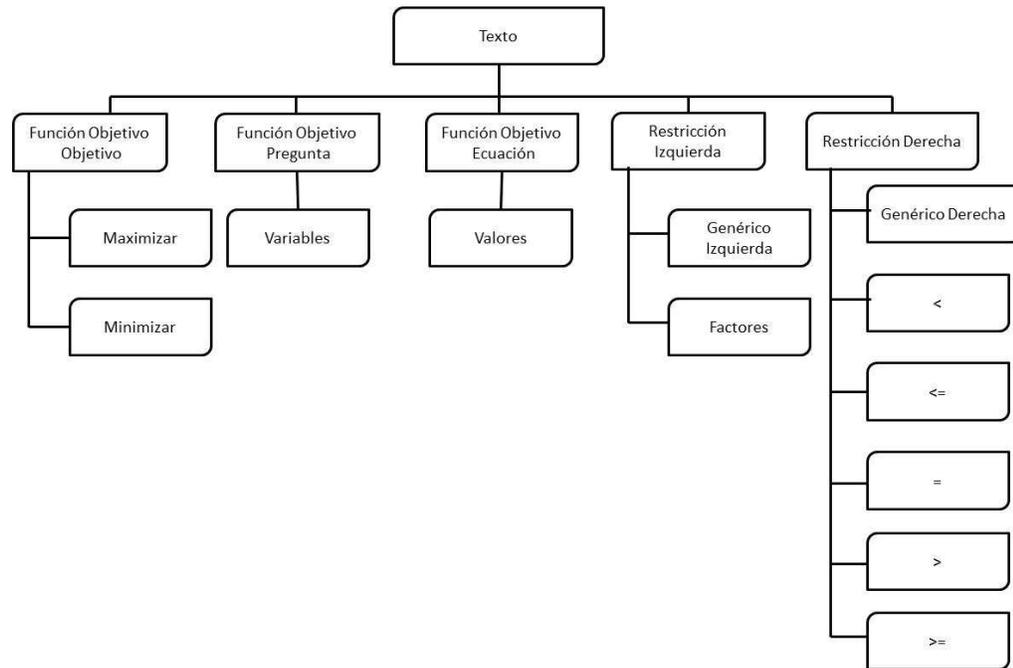
Figura 16 - Diagrama de Paquetes TesisIntento2



4.2.3 Resultados obtenidos: Los resultados obtenidos fueron los siguientes:

- Se logró que Dionisio determinara el objetivo de cada problema.
- Se construyó una base de datos con un mayor número de palabras y expresiones pertenecientes al léxico de Investigación de Operaciones, que fuesen susceptibles de ser utilizadas como los primeros patrones para la búsqueda de variables y recursos.
- Se estableció una jerarquía de clasificación con base en los componentes de un problema de simplex de programación lineal para que Dionisio llevara a cabo una mejor clasificación de subpárrafos. Esta jerarquía se estableció así (Figura 17):

Figura 17 - Jerarquía de Clasificación Inicial



- Se logró que Dionisio obtuviera elementos del modelado a través de algoritmos de búsqueda de variables y recursos.

Conclusiones

Después de realizado el segundo intento se concluyó que:

- La información dentro de los textos de un problema no puede ser extraída únicamente a través del reconocimiento de una palabra o expresión suelta.
- Los dos elementos fundamentales que Dionisio debe identificar son las variables y los recursos.
- Para lograr una buena clasificación se necesita primero una buena división del texto.

Trabajo posterior: Se identificó la necesidad de mejorar cada vez más los algoritmos de búsqueda de variables y recursos, la inserción de nuevos patrones de reconocimiento y un mejor tratamiento de la información con el ánimo de eliminar información irrelevante con la que Dionisio se confunde. Para ello se debe enriquecer cada vez más la base de datos y crear o dividir la información existente en nuevas tablas para un mejor tratamiento.

4.3 TESISINTENTO3

Después de establecer como prioridad de Dionisio la identificación de las variables y los recursos existentes en la redacción de los problemas de Simplex, en este intento se mejoran los algoritmos creados para la identificación e implementados en el intento anterior. Además de replantear los algoritmos de separación de subpárrafos y su clasificación, también se modifica la forma de manipular la información para que todo sea más fácil en términos de programación. Por otro lado, se realizan los primeros intentos de construcción del modelo, utilizando una matriz.

Se continúa con la alimentación y estructuración de la base de datos con el fin de que los procesos que lleva a cabo Dionisio puedan realizarse con mayor facilidad en cuanto al acceso a la información.

4.3.1 Análisis

Objetivos

- Hacer más fácil el manejo de la información a nivel de código.
- Eliminar los datos irrelevantes que Dionisio lista al tratar de obtener una variable o un recurso.
- Mejorar la división de los párrafos.
- Llegar hasta la construcción de un modelo matemático con un ejercicio determinado.
- Establecer una estructura que permita mostrar el modelo matemático por primera vez.

Estos objetivos se alcanzaran, a través de la adición de los módulos descritos a continuación a los ya construidos en los intentos anteriores.

Diagrama de casos de uso

Diagrama general: En este intento las acciones llevadas a cabo por Dionisio son básicamente las mismas del intento anterior, las diferencias se presentan a nivel de la forma como se maneja la información, como se muestra a continuación, en la Figura 18, la única nueva acción que aparece en este intento es la limpieza de los elementos del modelo, que para este intento es todavía muy elemental.

Figura 18 - Diagrama de Casos de Uso General TesisIntento3

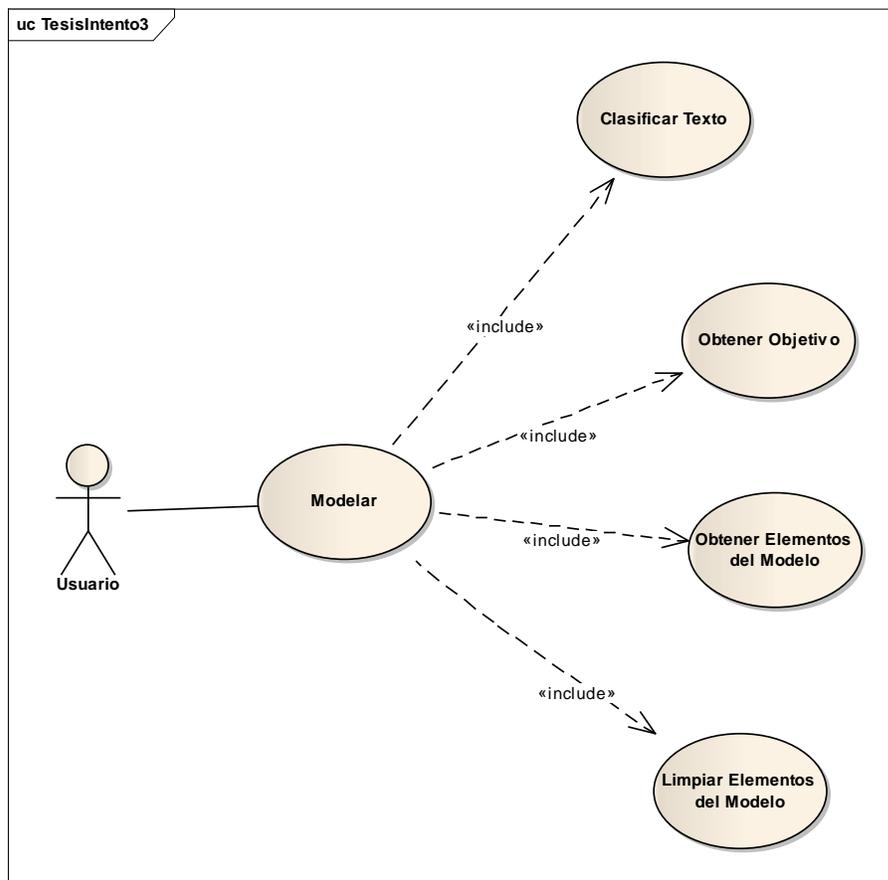


Diagrama específico: En el siguiente diagrama (Figura 19), se muestra en detalle las diferencias en cuanto al manejo de las estructuras, básicamente se puede ver que los elementos del modelo no se utilizan como simples cadenas de texto, en lugar de esto, se empiezan a armar estructuras a partir de los datos que estos elementos proveen.

Figura 19 - Diagrama de Casos de Uso TesisIntento3

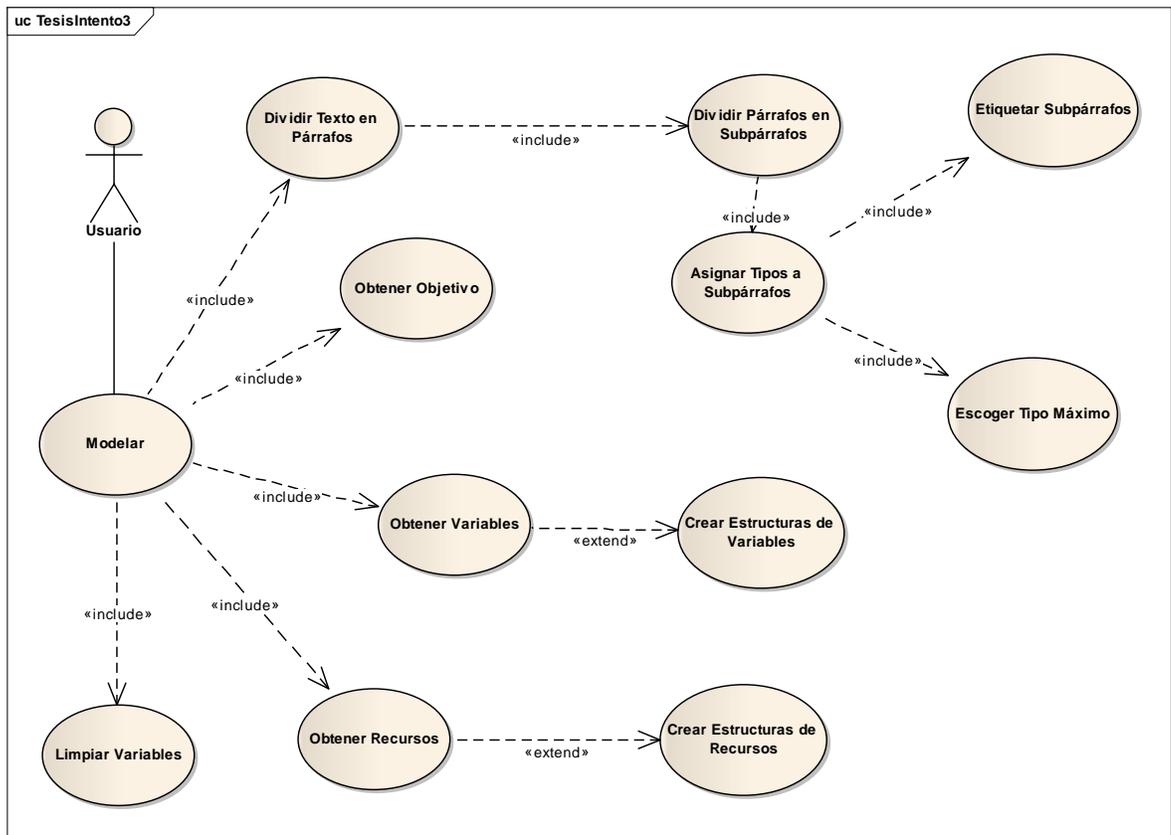
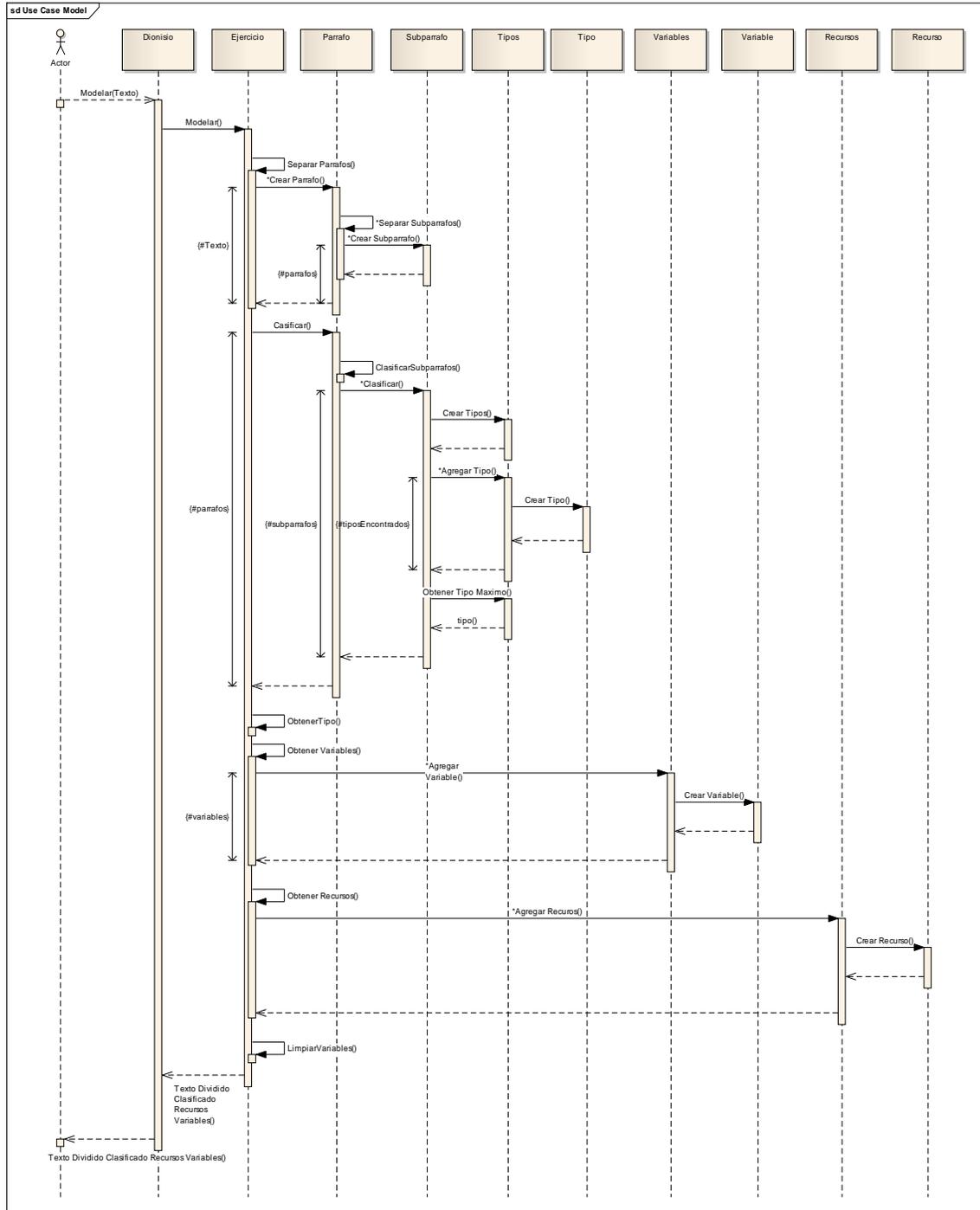


Diagrama de secuencia: En el siguiente diagrama (Figura 20), se evidencia que las variables se limpian al final, después de ser obtenidas.

Figura 20 - Diagrama de Secuencia TesisIntento3



4.3.2 Diseño

Diagrama general de datos: En este intento se suman nuevas tablas que almacenan y relacionan datos con el fin de realizar una mejor división del texto, a continuación un diagrama general de la base de datos utilizada por Dionisio en este intento (Figura 21).

Figura 21 - Diagrama General de Datos TesisIntento3



Diagrama conceptual: En este intento se evidenció la necesidad de organizar las abreviaciones, para así saber, si en un párrafo se habla de una unidad de medida, ya sea tiempo, volumen, área, etc. esto, con el fin de establecer el lugar en el texto, dónde buscar los recursos y las restricciones (Figura 22).

Figura 22 - Diagrama Conceptual TesisIntento3

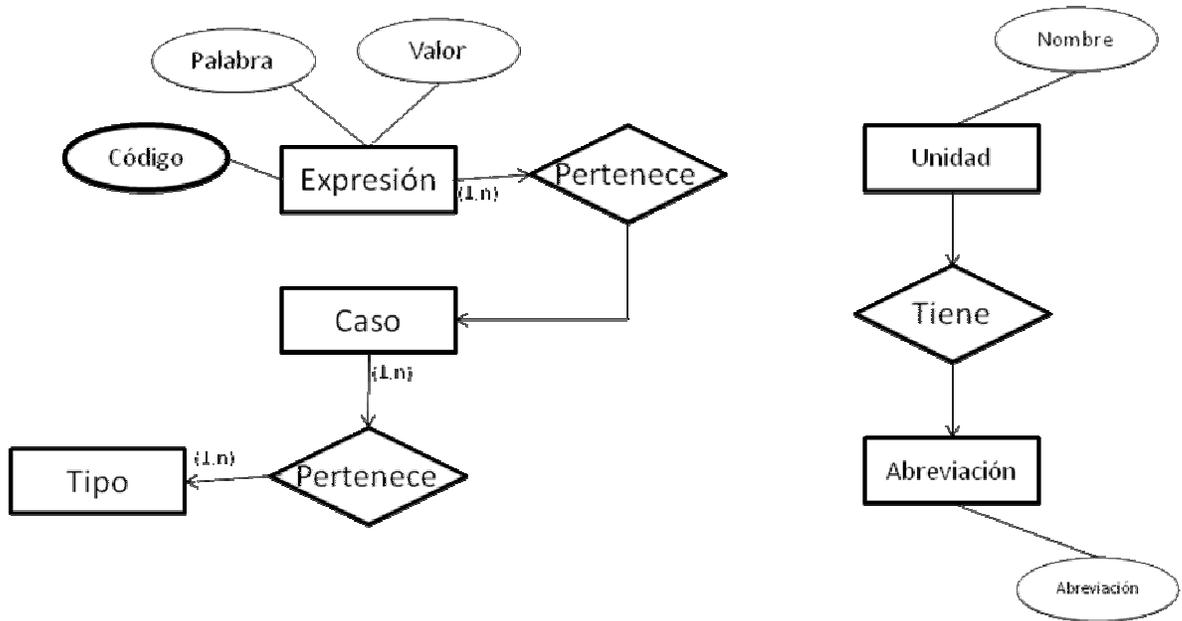


Diagrama de BD: Estas nuevas tablas y las existentes desde intentos previos, están estructuradas como se muestra a continuación (Figura 23).

Figura 23 - Diagrama de Base de Datos TesisIntento3

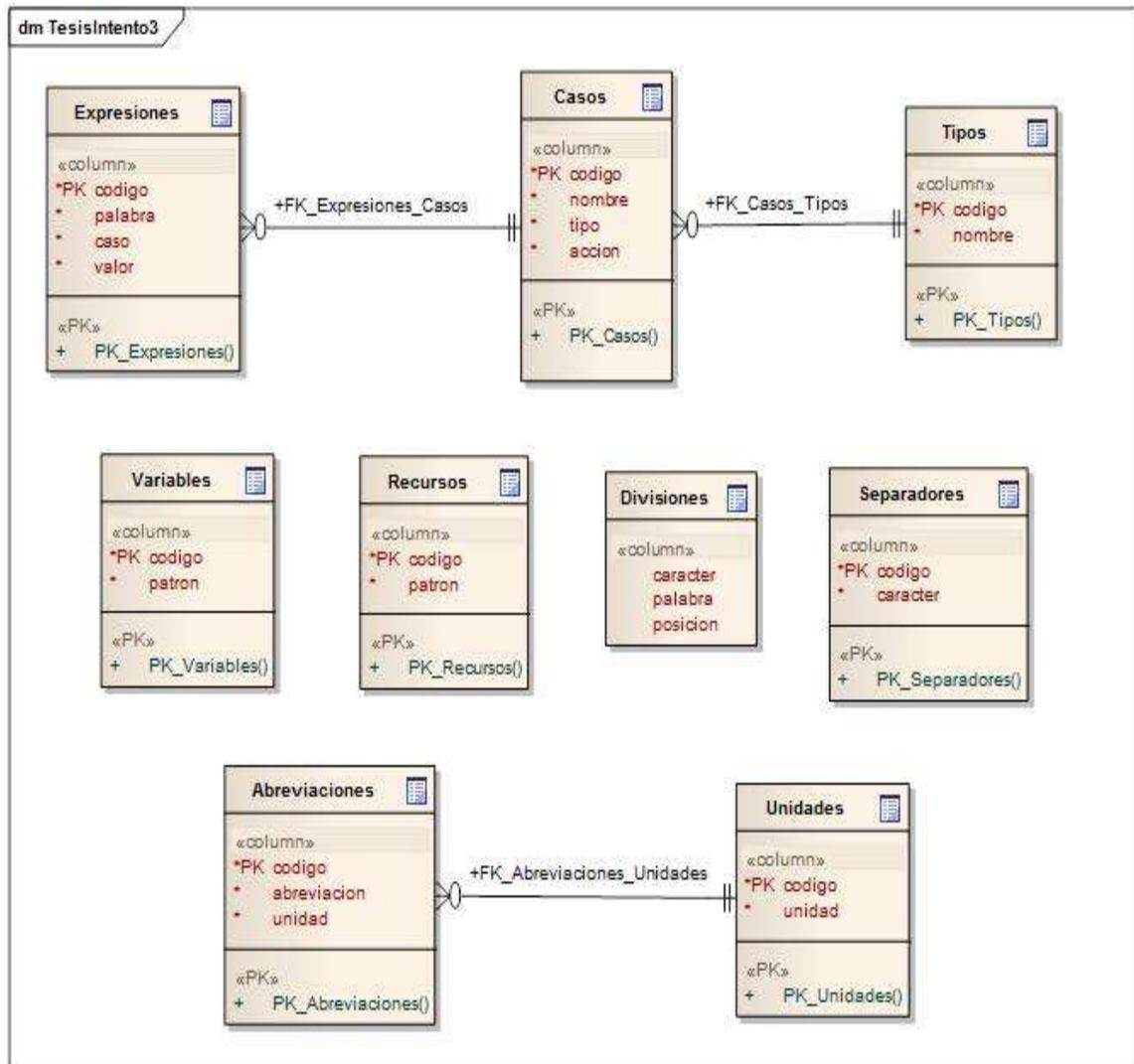


Diagrama de clases: En este intento se organiza la estructura de Dionisio para separar las funciones del manejo del ejercicio y las de manejo de variables y recursos (Figura 24).

Figura 24 - Diagrama de Clases TesisIntento3

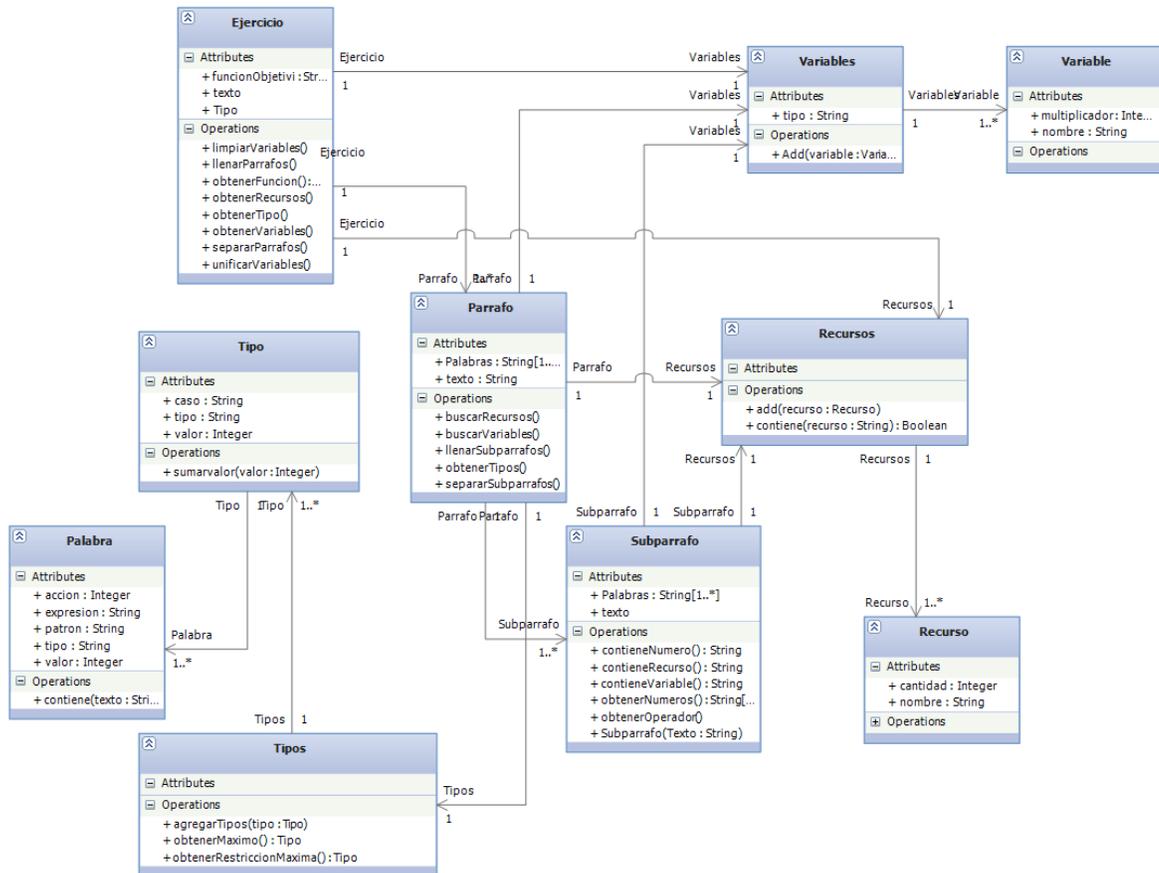
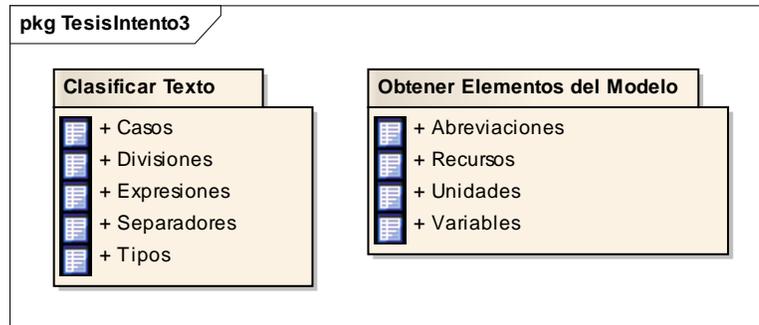


Diagrama de paquetes: En este intento, las funcionalidades que se implementaron en el funcionamiento de Dionisio, son las mismas del segundo intento, la diferencia como se presenta en el siguiente diagrama, radica en que ahora estos módulos utilizan las nuevas tablas que aparecen en este intento (Figura 25).

Figura 25 - Diagrama de Paquetes TesisIntento3



4.3.3 Resultados obtenidos: En este intento se obtuvieron los siguientes resultados:

- Creación de estructuras de información para variables y recursos que permitan almacenar más información de cada uno de estos elementos además de facilitar su manipulación a nivel de código.
- Primeros algoritmos de limpieza y unificación de variables y recursos a través de los cuales se depuran los listados resultantes de las búsquedas de variables y recursos realizadas por Dionisio.
- Modificación de algoritmos de división de texto, para hacerlos más eficientes.
- Creación de una primera representación del modelo, a través de la siguiente matriz (Figura 26):

Figura 26 - Matriz de Modelado Inicial

	v1	v2	OPERADOR	VALOR
r1	n11	n21	</>/<=>=</>=	l1
r2	n12	n22	</>/<=>=</>=	l2
r3	n13	n23	</>/<=>=</>=	l3
F.OBJETIVO	f1	f2		Max/Min

En esta matriz, las filas r1, r2, r3 representan los recursos con sus respectivos límites l1, l2 y l3, mientras que las columnas v1 y v2 representan las variables.

- Establecimiento de las reglas básicas para la construcción de un modelo. El resultado fue el siguiente:
 - Si en un subpárrafo se encuentra un número, una variable y un recurso, este número es la cantidad de recurso que consume la variable. Dentro del modelo hace parte de la inecuación o restricción del recurso.
 - Si en un subpárrafo se encuentra un número, una palabra que restrinja y un recurso, este número es la cantidad del recurso restringida por la palabra o el límite del recurso.
 - Si en un subpárrafo se encuentra un número y una variable acompañados de palabras propias de función objetivo, este número es el aporte de la variable a la función objetivo.

Conclusiones

- La identificación de recursos es más complicada que la identificación de variables; se acumula más basura difícil de eliminar y los patrones no son tan genéricos como los correspondientes a las variables.
- La división del texto y más específicamente la división de párrafos en subpárrafos no siempre se ajusta a los signos de puntuación y conectores.
- Las variables, por lo regular, tienen un tipo que es conveniente identificar porque algunas restricciones vienen dadas según este tipo.

- Las variables y los recursos no son simples cadenas de texto. Variables y recursos tienen en su contexto información importante del modelo, por lo que es necesario utilizar estructuras o clases especiales para manejarlas.
- El proceso de construcción del modelo puede llevarse a cabo exitosamente a través de reglas simples, siempre y cuando el paso previo de clasificación esté bien construido.

Trabajo posterior: Como trabajo futuro siempre queda seguir alimentando la base de datos y mejorando el proceso de la identificación de recursos y variables, además de su limpieza. Paralelamente a estos trabajos se está en un continuo pensamiento del problema con el fin de idear nuevos métodos que faciliten los procesos y que hagan que Dionisio mejore. Esta vez la idea que justifica continuar al siguiente intento es que se encontró una nueva jerarquía que se estaba obviando en el momento de la clasificación.

4.4 TESISINTENTO4

Este intento se basa en los adelantos que se lograron en el tercer intento. Ahora se continúa perfeccionando la búsqueda de variables y recursos ya que se definieron estos dos elementos como fundamentales en la construcción del modelo. Para ello ahora esa búsqueda se realizara a nivel de párrafo y dependiendo de la clasificación del párrafo, ya que anteriormente se realizaba en todo el texto sin darle la debida atención a la clasificación.

4.4.1 Análisis

Objetivos

- Como la clasificación de texto toma un lugar importante dentro del proceso de Dionisio es necesario perfeccionarla.
- Empezar a solucionar la ambigüedad presente en el significado de las expresiones.

En este intento, y con el fin de alcanzar los objetivos propuestos, se plantea adicionar a Dionisio, los módulos, como se describe a continuación:

Diagrama de casos de uso

Diagrama general: A continuación (Figura 27), se presentan las acciones necesarias para que Dionisio cumpla los objetivos planteados, este diagrama no varía con respecto al anterior, ya que la diferencia está a nivel de algoritmos en cuanto a la información que se utiliza para clasificar el texto basándose en la nueva jerarquía.

Figura 27 - Diagrama de Casos de Uso General TesisIntento4

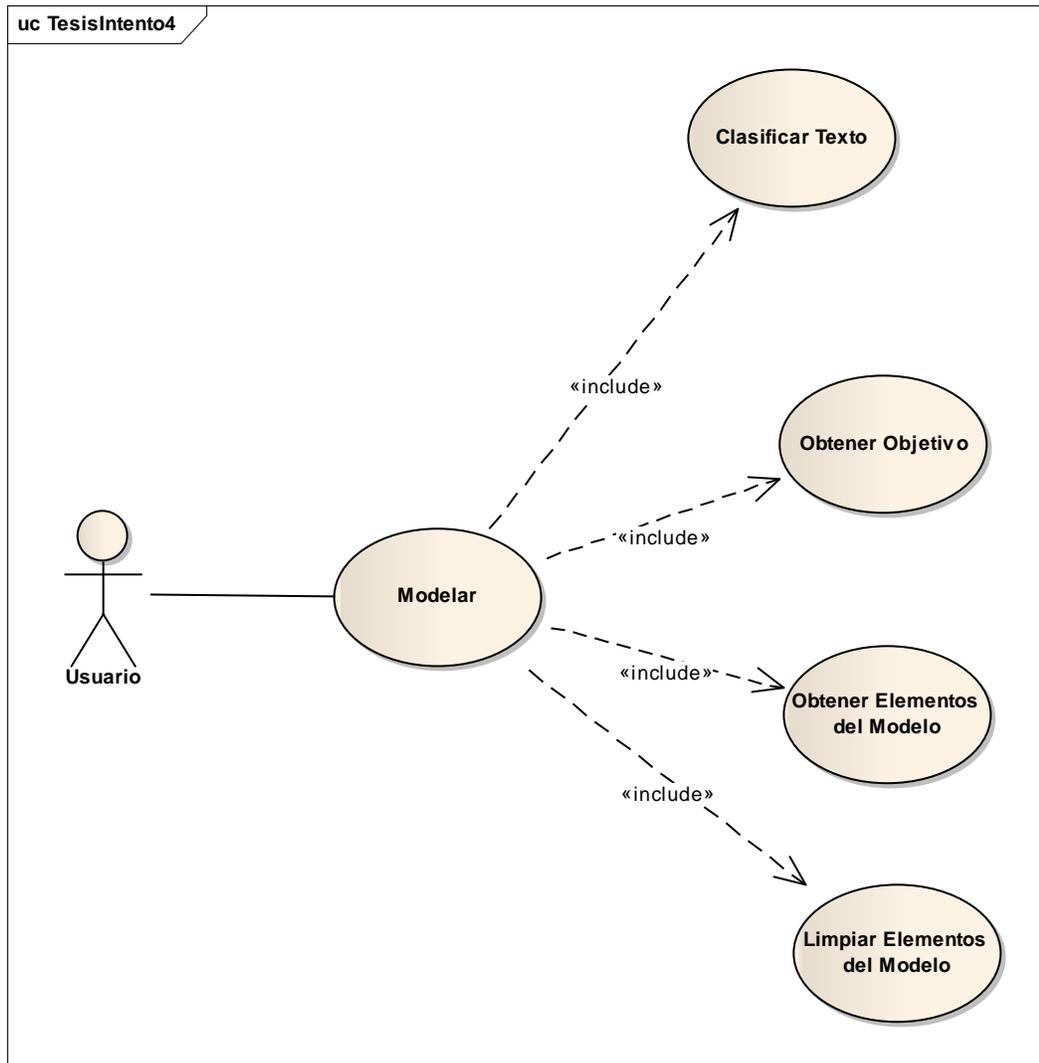


Diagrama específico: En el siguiente diagrama (Figura 28), se presentan con mayor detalle las acciones realizadas por Dionisio, esta vez no varía en nada con respecto al del tercer intento.

Figura 28 - Diagrama de Casos de Uso TesisIntento4

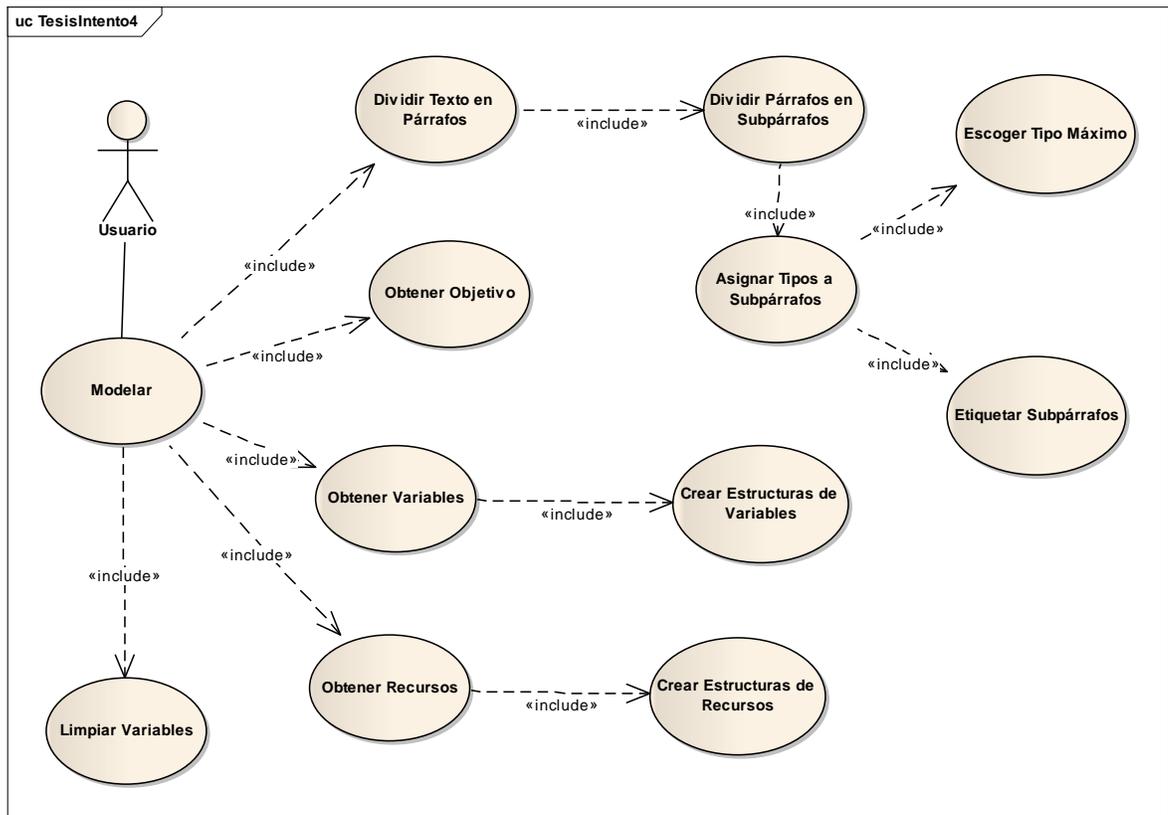
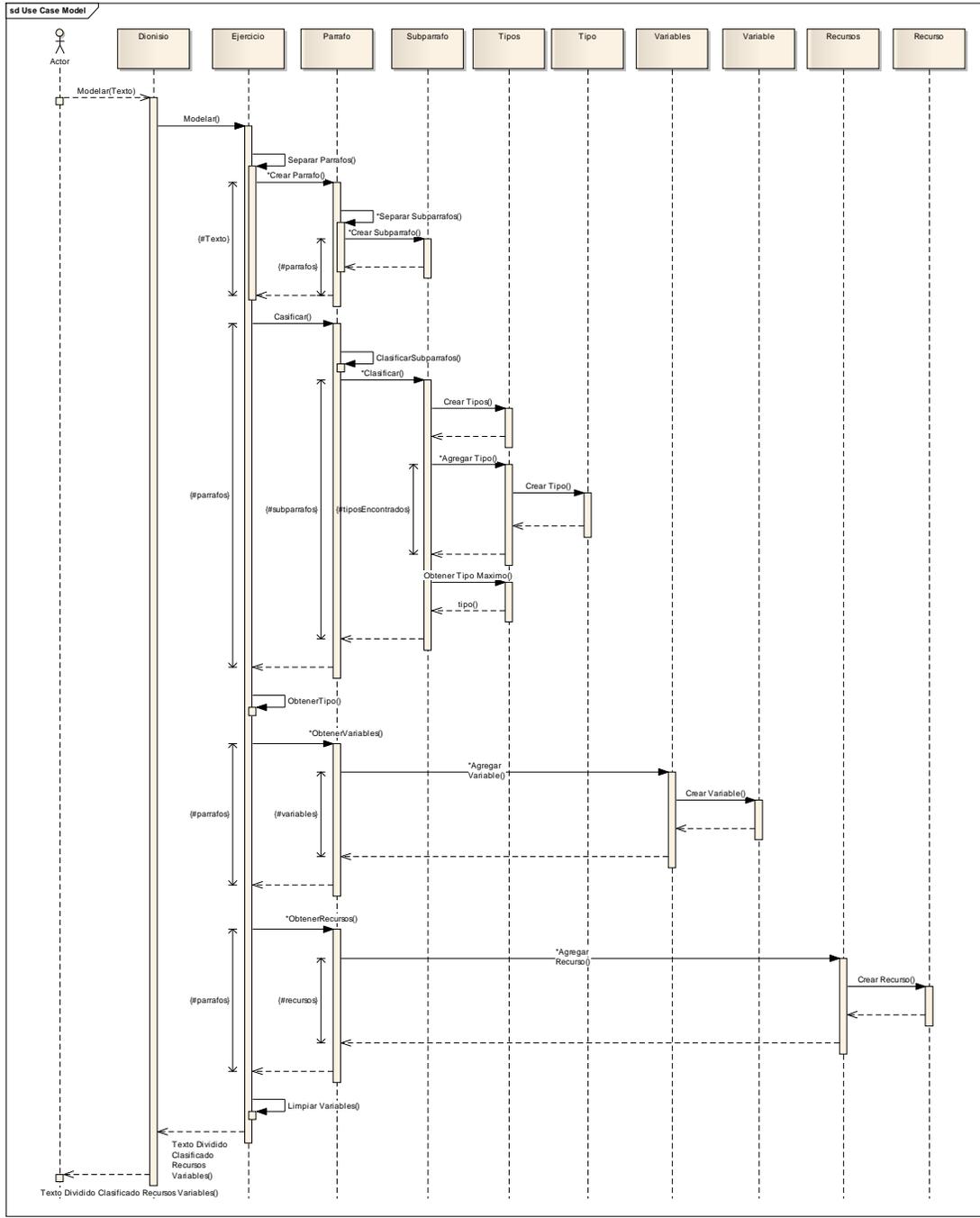


Diagrama de secuencia

Figura 29 - Diagrama de Casos de Uso TesisIntento4



4.4.2 Diseño

Diagrama general de datos: El gran cambio de este intento, es la modificación a la jerarquía que se venía utilizando para llevar a cabo la clasificación del texto, como se muestra a continuación (Figura 30). En este intento se anexa una nueva tabla, llamada generales la cual almacena los datos del nuevo nivel de la jerarquía.

Figura 30 - Diagrama General de Datos TesisIntento4



Diagrama conceptual: En el siguiente diagrama (Figura 31), se evidencia como un tipo ya pertenece a un tipo más general, dentro de la jerarquía de clasificación.

Figura 31 - Diagrama Conceptual TesisIntento4

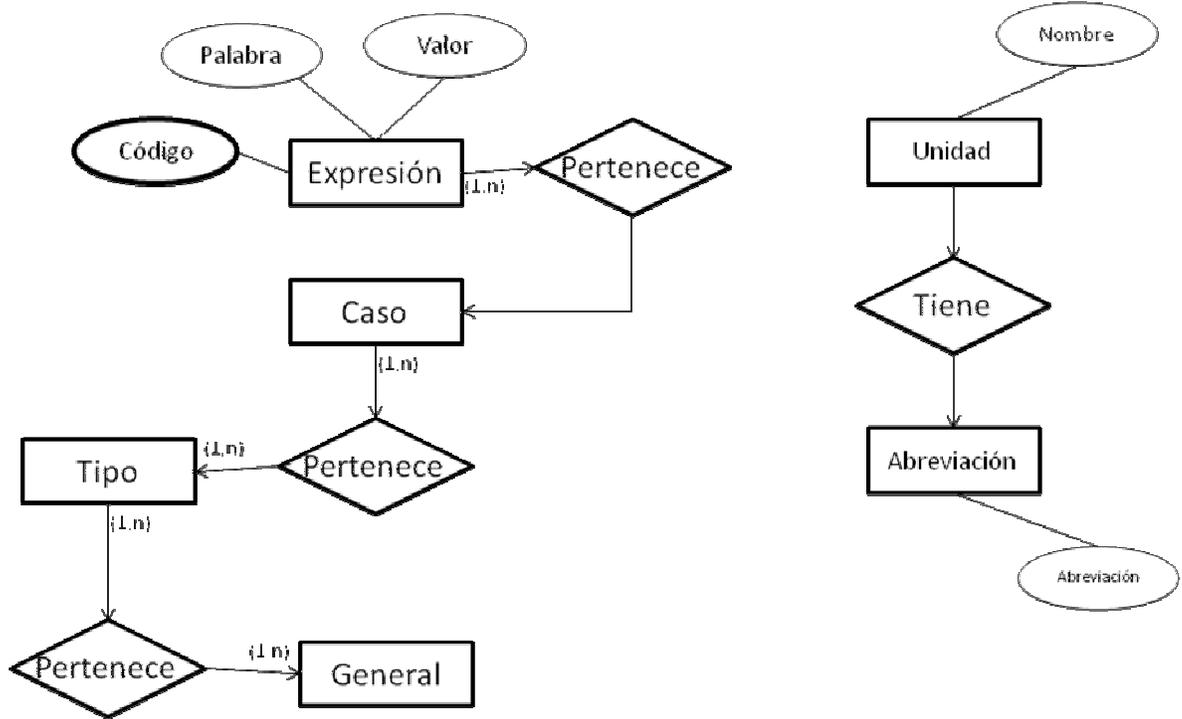


Diagrama de BD: A continuación (Figura 32) se presenta la nueva estructura de la base de datos, con la nueva tabla que modifica la jerarquía de clasificación.

Figura 32 - Diagrama de Base de Datos TesisIntento4

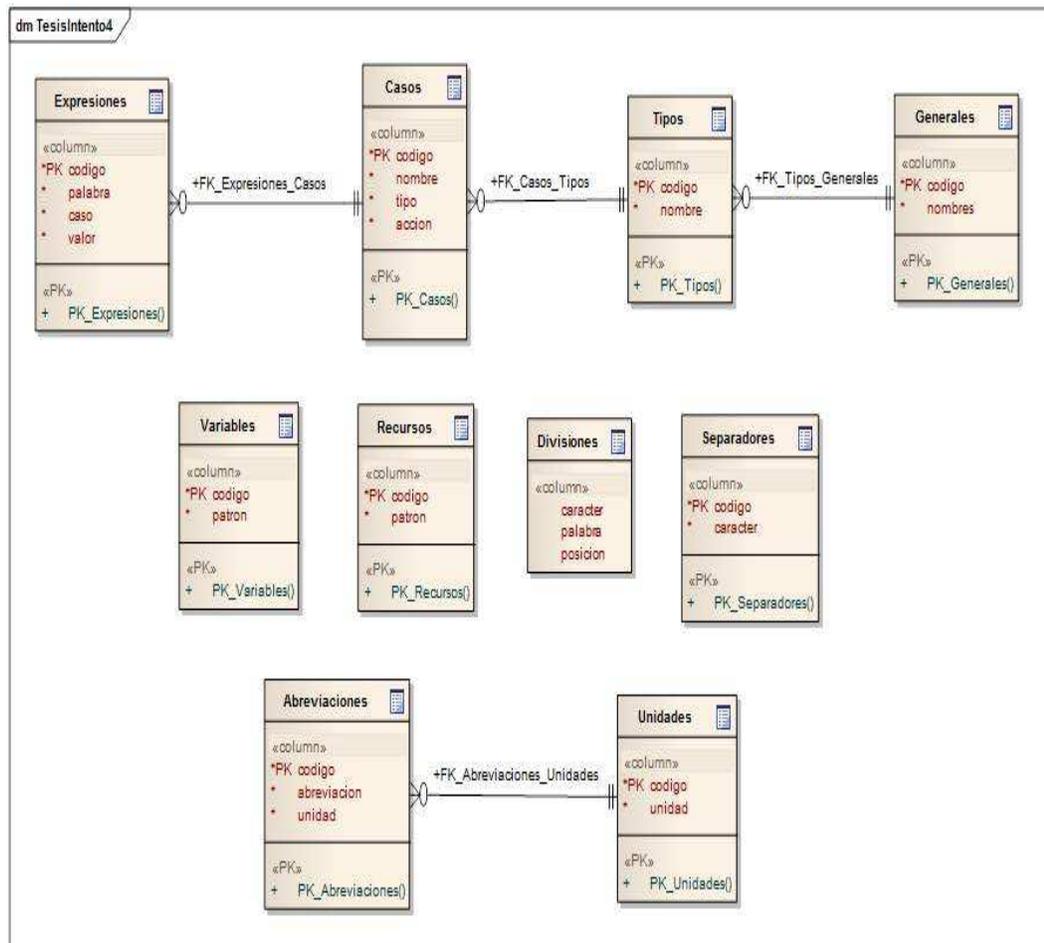


Diagrama de clases: Aunque en la base de datos fue creada la nueva tabla, para el nuevo nivel de jerarquía, en este intento no se consideró necesaria la creación de una clase para su manejo ver Figura 33.

Figura 33 - Diagrama de Clases TesisIntento4

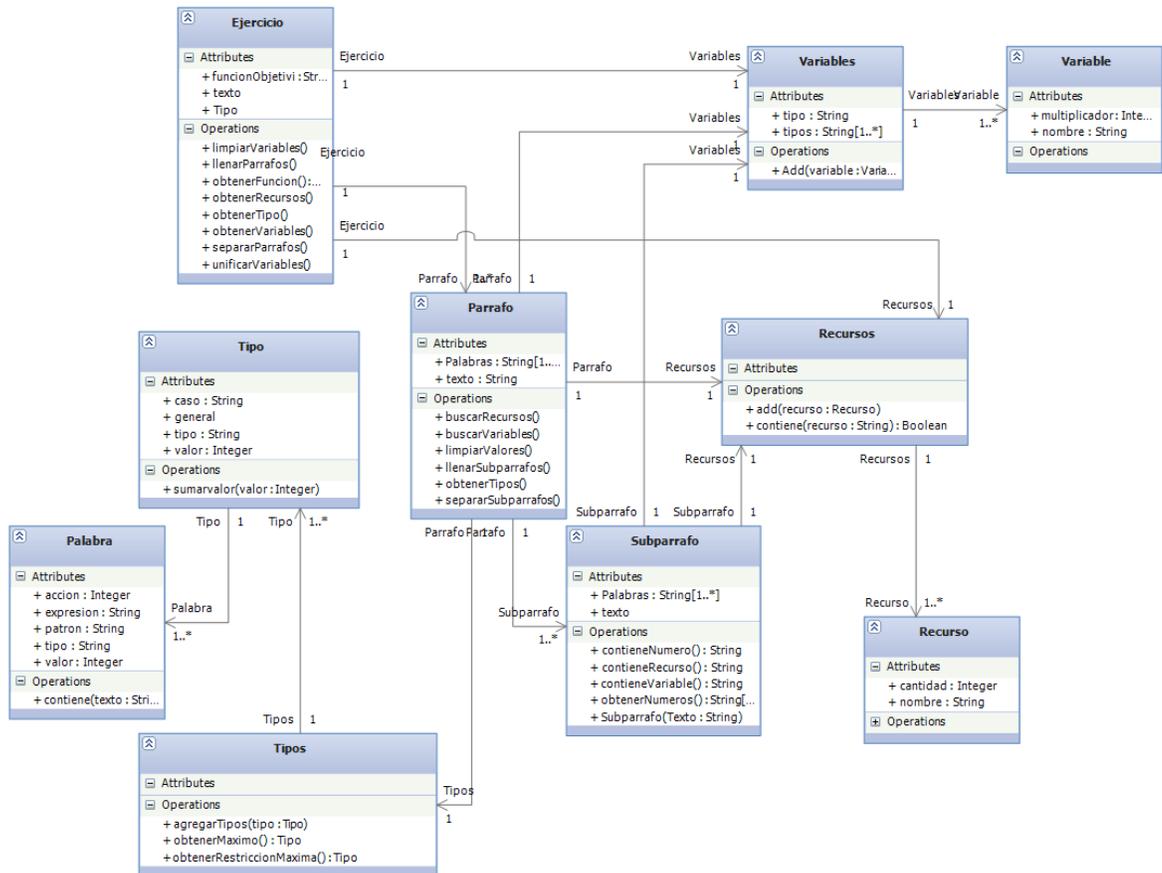
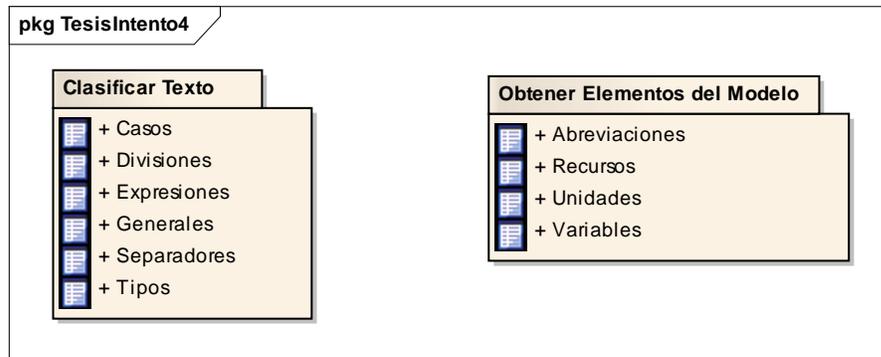


Diagrama de paquetes: Las funcionalidades de Dionisio, continúan siendo las mismas del anterior intento, ahora, como se muestra en el siguiente diagrama (Figura 34), la funcionalidad de clasificar texto, utiliza la nueva tabla generales.

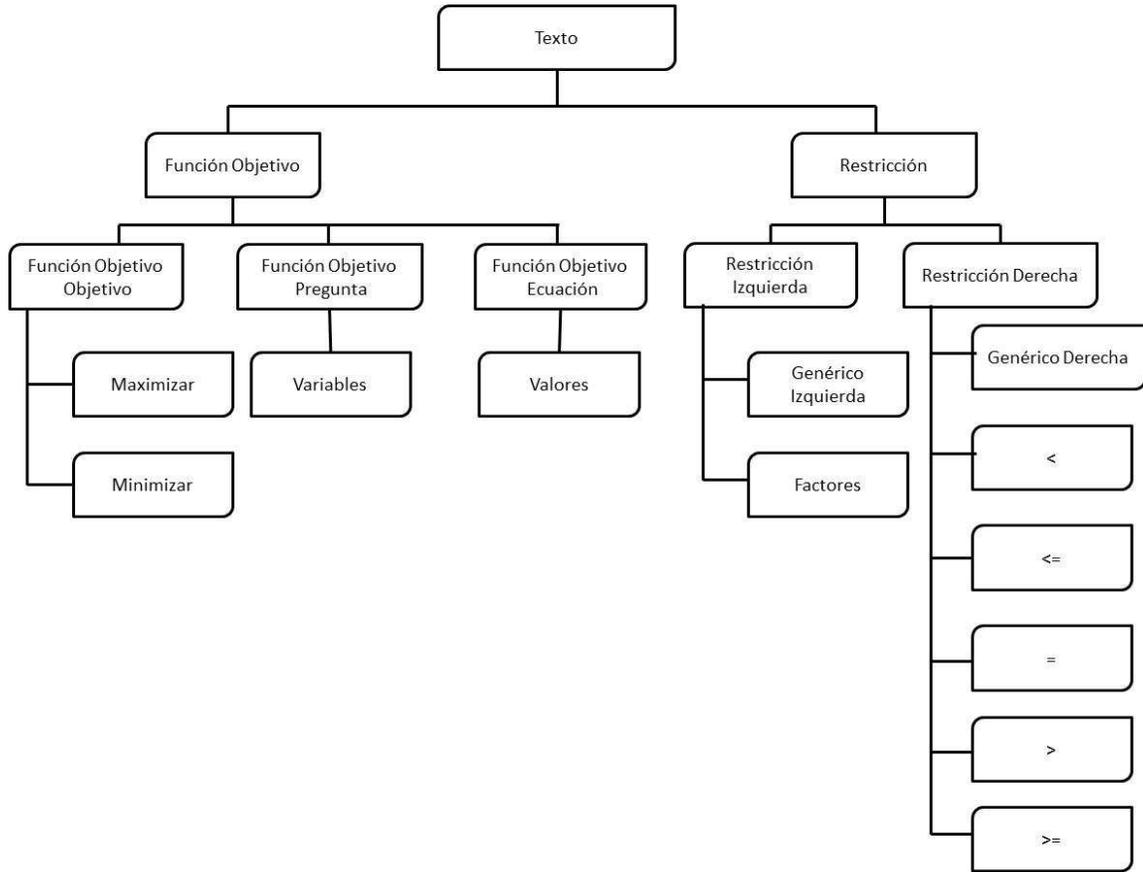
Figura 34 - Diagrama de Paquetes TesisIntento4



4.4.3 Resultados obtenidos: Realizado el trabajo se obtienen los siguientes resultados:

- Se realizaron unos primeros avances al tratar la ambigüedad de las expresiones. Se encontró que una palabra o expresión cambia el significado dependiendo del objetivo del problema; lo mismo sucede con los patrones utilizados en la búsqueda.
- Nueva jerarquía de clasificación (Figura 35), la cual será definitiva para el resto del desarrollo de Dionisio. Ésta es:

Figura 35 - Jerarquía de Clasificación Final



Esta nueva jerarquía tiene una única variación con respecto a la que se estaba utilizando: se añadieron dos nuevas clasificaciones que agrupan de mejor manera a las más específicas. Estas son Función Objetivo y Restricción que, aunque se sabía que existían y se utilizaban, se obviaban en la anterior jerarquía.

Conclusiones

- La clasificación del texto es la clave para lograr una buena obtención de elementos del modelo y la futura construcción del mismo.
- Las palabras, expresiones o patrones que se apliquen en el proceso de clasificación y búsqueda dependen del objetivo del problema.

Trabajo posterior: La obtención de los recursos sigue siendo el obstáculo que debe superarse. Por ello, los siguientes intentos se centrarán en obtener estos elementos del modelo, sin descuidar la continua alimentación y estructuración de la base de datos.

4.5 TESISINTENTOS

Debido a que, hasta el momento, la identificación de recursos ha sido uno de los mayores obstáculos que ralentizan el proyecto, se decide centrar la atención en los algoritmos que los buscan y más específicamente en los de limpieza y unificación.

4.5.1 Análisis

Objetivos

- Obtención de elementos del modelo sin basura o con la máxima limpieza posible, poniendo especial atención en la obtención de los recursos.
- Organización de elementos del código que están sueltos hasta ahora.

Para lograr esto se propusieron los siguientes diagramas:

Diagrama de casos de uso

Diagrama general: Como se mencionó anteriormente, en este intento se añade una nueva acción llamada Unificar Elementos, con el fin de depurar la información abstraída por Dionisio, en el siguiente diagrama (Figura 36), se ilustra esta nueva acción, junto a las acciones implementadas en intentos anteriores.

Figura 36 - Diagrama de Casos de Uso General TesisIntento5

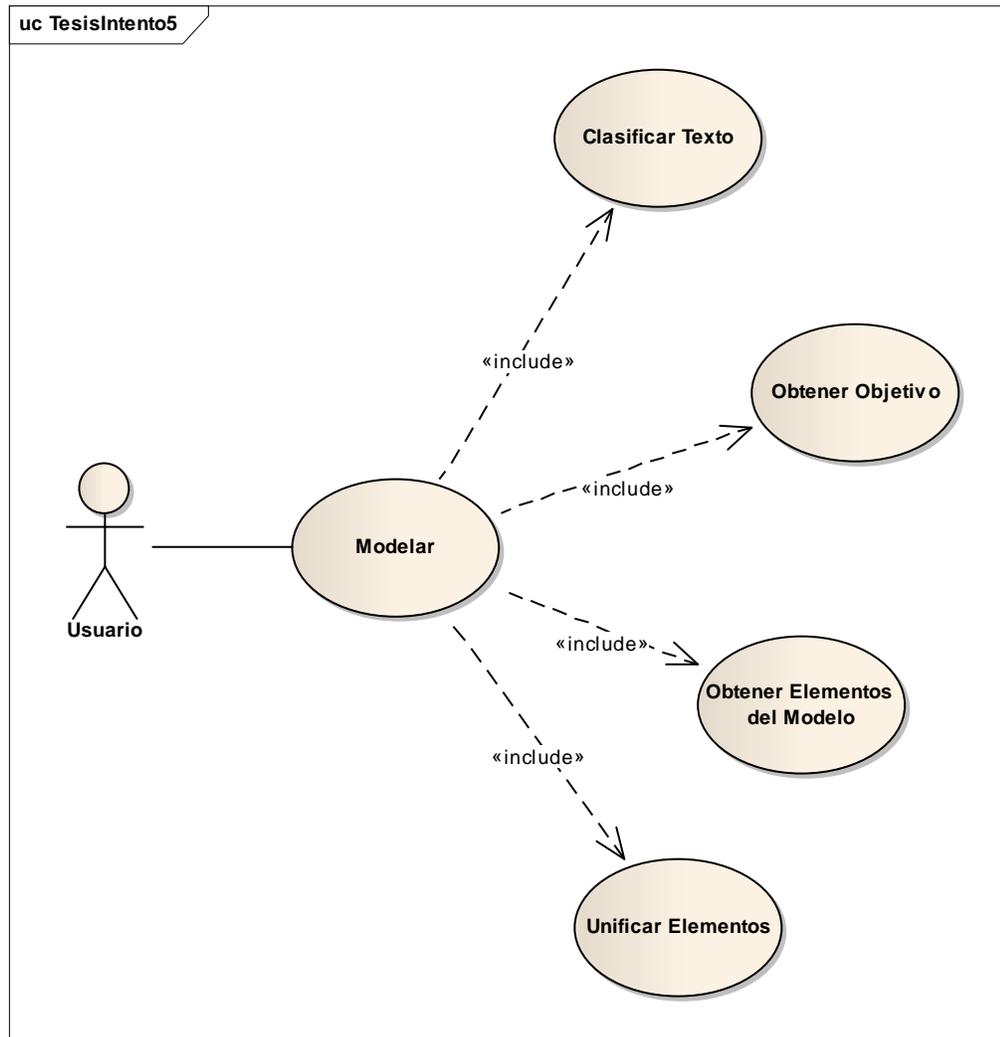


Diagrama específico: A continuación (Figura 37) se muestra las nuevas funcionalidades junto con las creadas en intentos anteriores, de una forma más detallada.

Figura 37 - Diagrama de Casos de Uso TesisIntento5

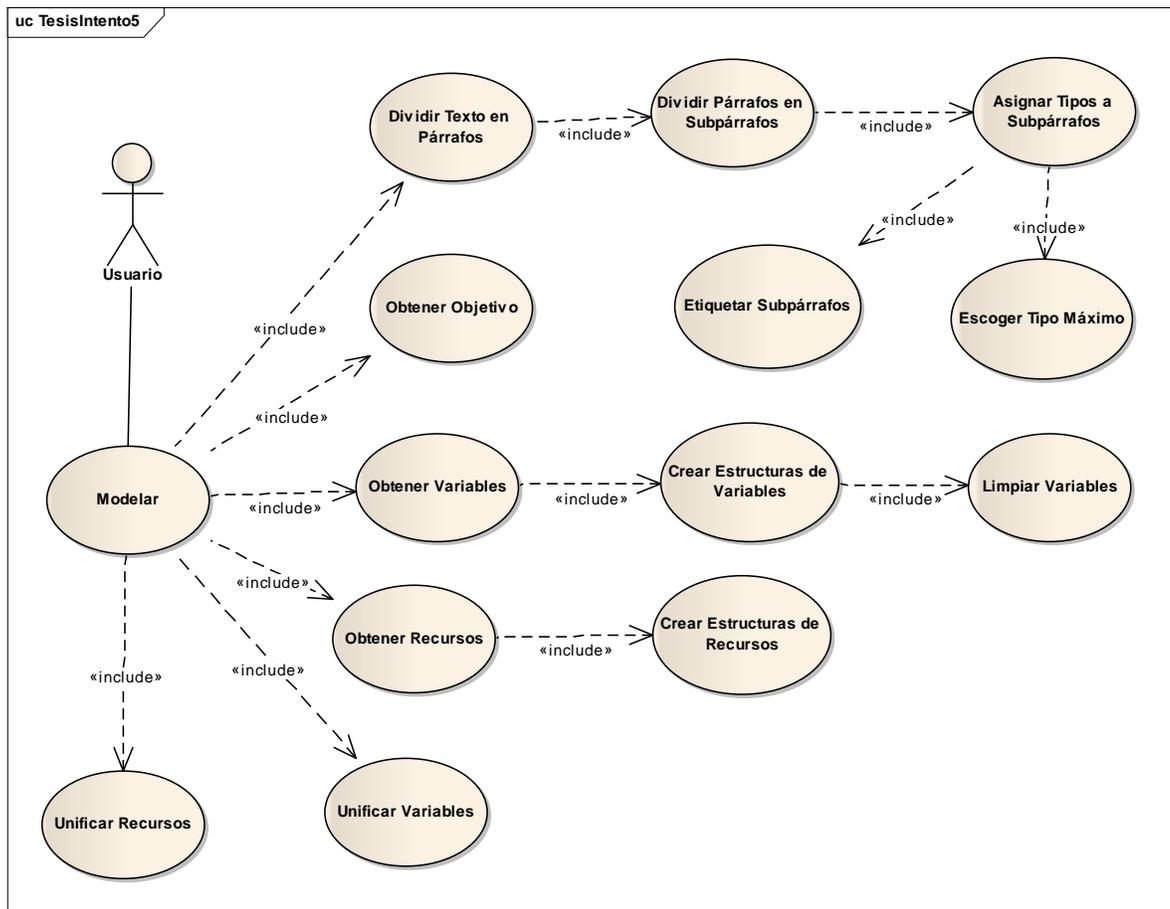
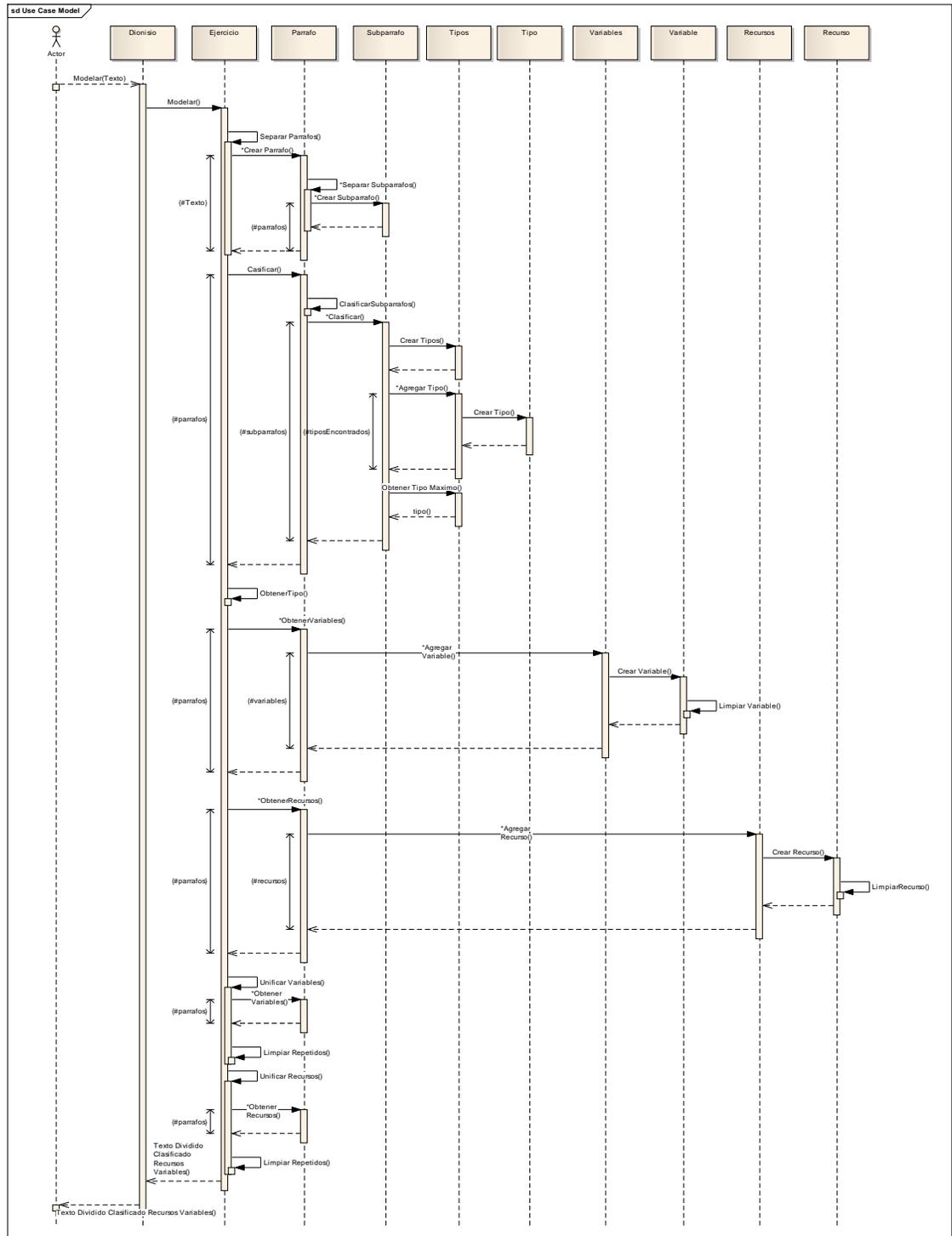


Diagrama de secuencia: En este intento se estableció que las variables y los recursos serían limpiados conforme son obtenidas. Ver Figura 38.

Figura 38 - Diagrama de Secuencia TesisIntento5



4.5.2 Diseño

Diagrama general de datos: En este intento a nivel de base de datos y en busca de lograr una obtención de elementos del modelo exitosa, se suma la tabla Quitar, donde se almacena información, que se debe eliminar de los listados de elementos obtenidos por Dionisio. Ver Figura 39.

Figura 39 - Diagrama General de Datos TesisIntento5



Diagrama conceptual

Figura 40 - Diagrama Conceptual TesisIntento5

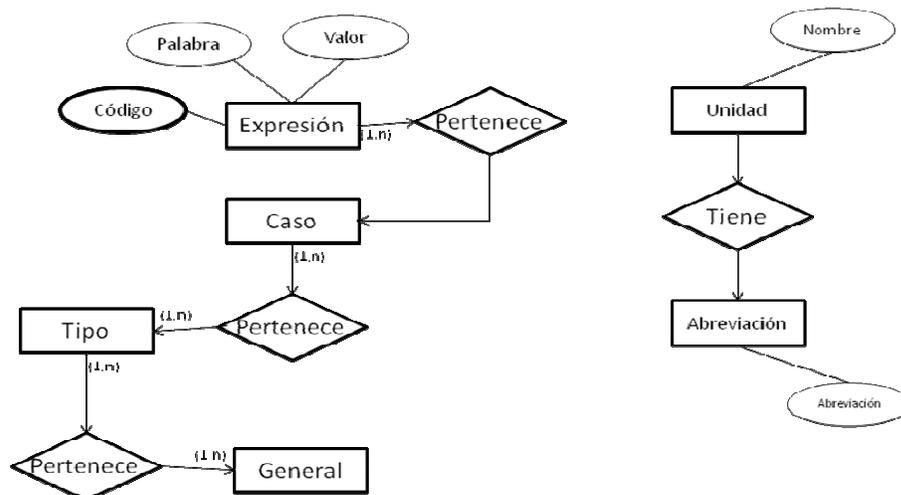


Diagrama de BD: La estructura de la base de datos utilizada por Dionisio en esta iteración, es la que se presenta en el siguiente diagrama (Figura 41).

Figura 41 - Diagrama de Base de Datos TesisIntento5

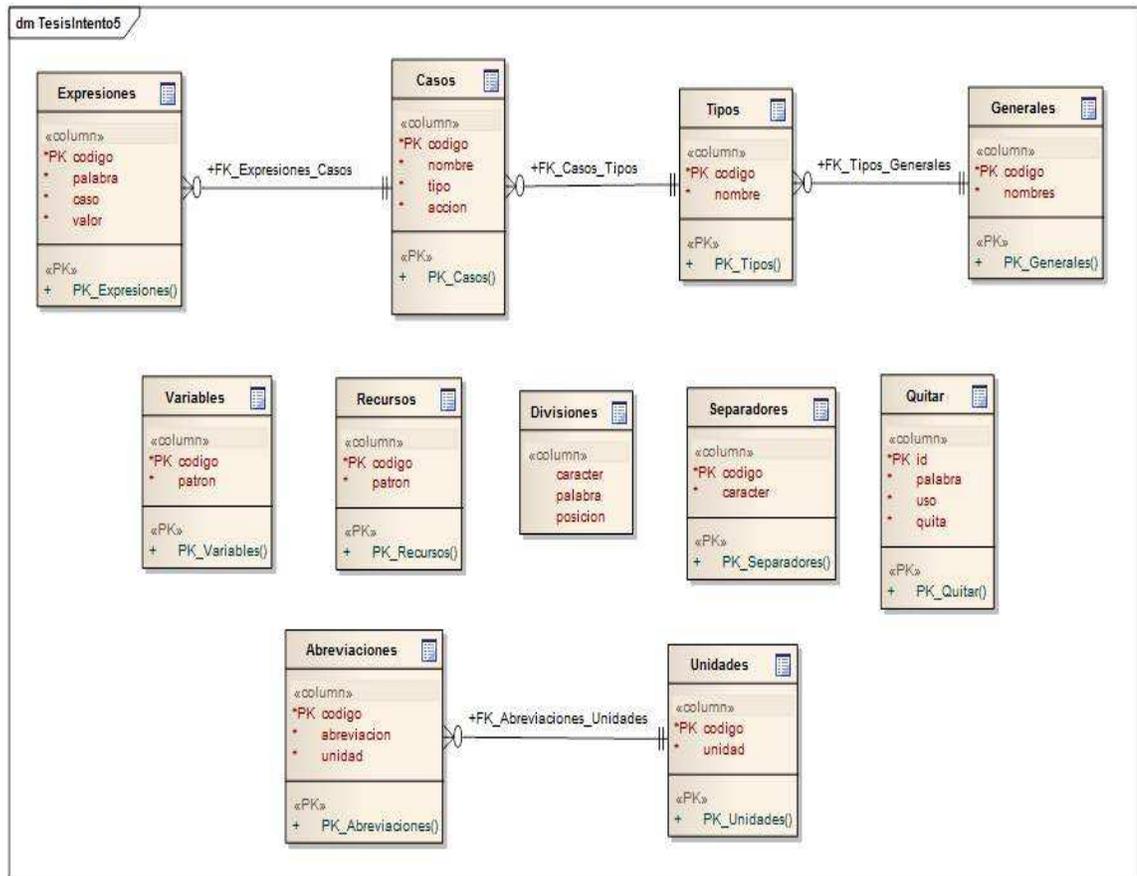


Diagrama de clases

Figura 42 - Diagrama de Clases TesisIntento5

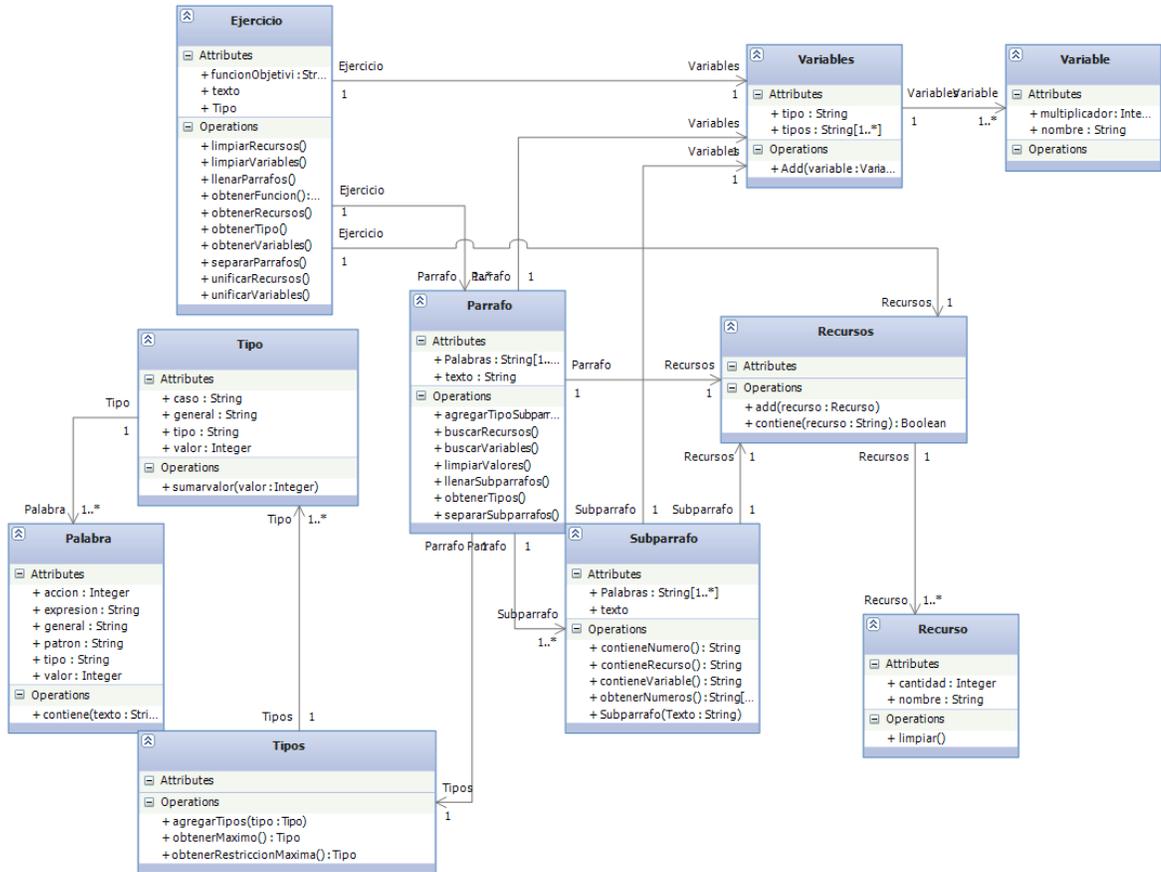
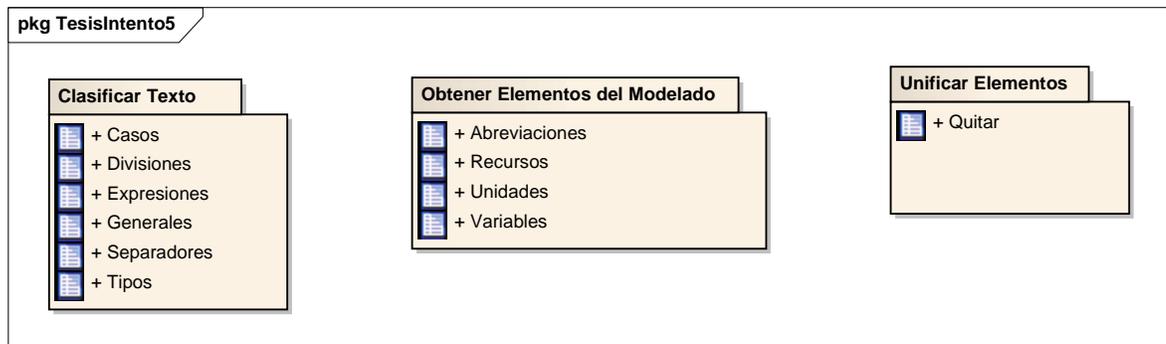


Diagrama de paquetes: En este intento, aparece una nueva funcionalidad denominada Unificar Elementos, que tiene como objetivo depurar los elementos que Dionisio identifica a partir de los dos primeros módulos, esta nueva funcionalidad a nivel de base de datos, utiliza la tabla Quitar, como se muestra a continuación.

Figura 43 - Diagrama de Paquetes TesisIntento5



4.5.3 Resultados obtenidos: Se organizaron mejor algunas partes del código que se estructuran en la base de datos. Los principales aportes de este intento son las ideas que surgieron. Éstas cambiaron significativamente el orden de los procesos dentro de Dionisio y causaron que el siguiente intento fuese el último.

Conclusiones

- En este momento se llegó a un punto del proyecto en el cual se hizo necesario replantear la forma en la que Dionisio está realizando el proceso de modelado. Los resultados no variaron significativamente con respecto al anterior.

Trabajo posterior: Se impone la necesidad de replantear el proceso de Dionisio y la continua alimentación de la base de datos.

4.6 TESISINTENTO6 (INTENTO FINAL)

Es el intento final de Dionisio. En esta iteración se replantea la forma en la que Dionisio está llevando a cabo el proceso de modelado, debido principalmente al estancamiento que originó la obtención de recursos ya que se producían muchos datos erróneos.

Antes de entrar a describir este intento en detalle, se lista la forma en que Dionisio realizaba el proceso. Básicamente se hacía a través de los siguientes nueve pasos generales:

1. Dionisio obtiene el objetivo.
2. Dionisio divide el texto.
- 3. *Dionisio clasifica las partes resultantes de la división del texto.***
- 4. *Dionisio busca variables en todo el texto.***
5. Dionisio limpia y unifica variables.
- 6. *Dionisio busca recursos en todo el texto.***
7. Dionisio limpia y unifica recursos.
8. Dionisio arma el modelo dependiendo de los elementos extraídos.
9. Dionisio presenta el modelo.

Los tres pasos resaltados deben ser replanteados para, con base en los nuevos resultados, atacar el séptimo paso, con el fin de lograr que Dionisio realice su principal objetivo.

4.6.1 Análisis

Objetivos

- Mejorar la clasificación de subpárrafos.
- Obtención correcta de elementos del modelado.
- Crear estructuras para construir el modelo.
- Hacer que Dionisio construya los modelos.

Los objetivos propuestos se alcanzaran construyendo los módulos que a continuación se presentan:

Diagrama de casos de uso

Diagrama general: Al final las acciones llevadas a cabo por Dionisio, son las que se presentan en el siguiente diagrama (Figura 44):

Figura 44 - Diagrama de Casos de Uso General TesisIntento6

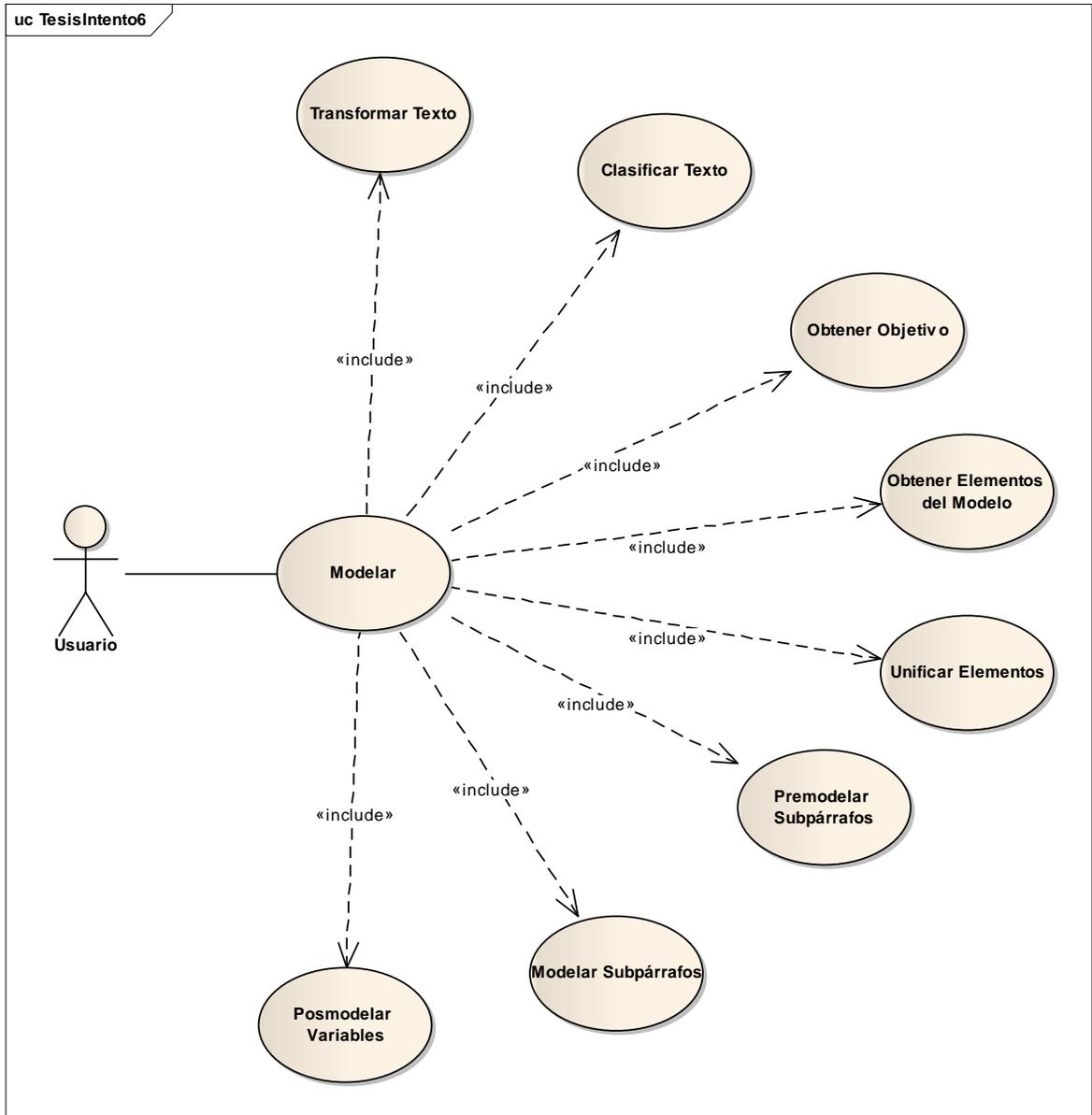
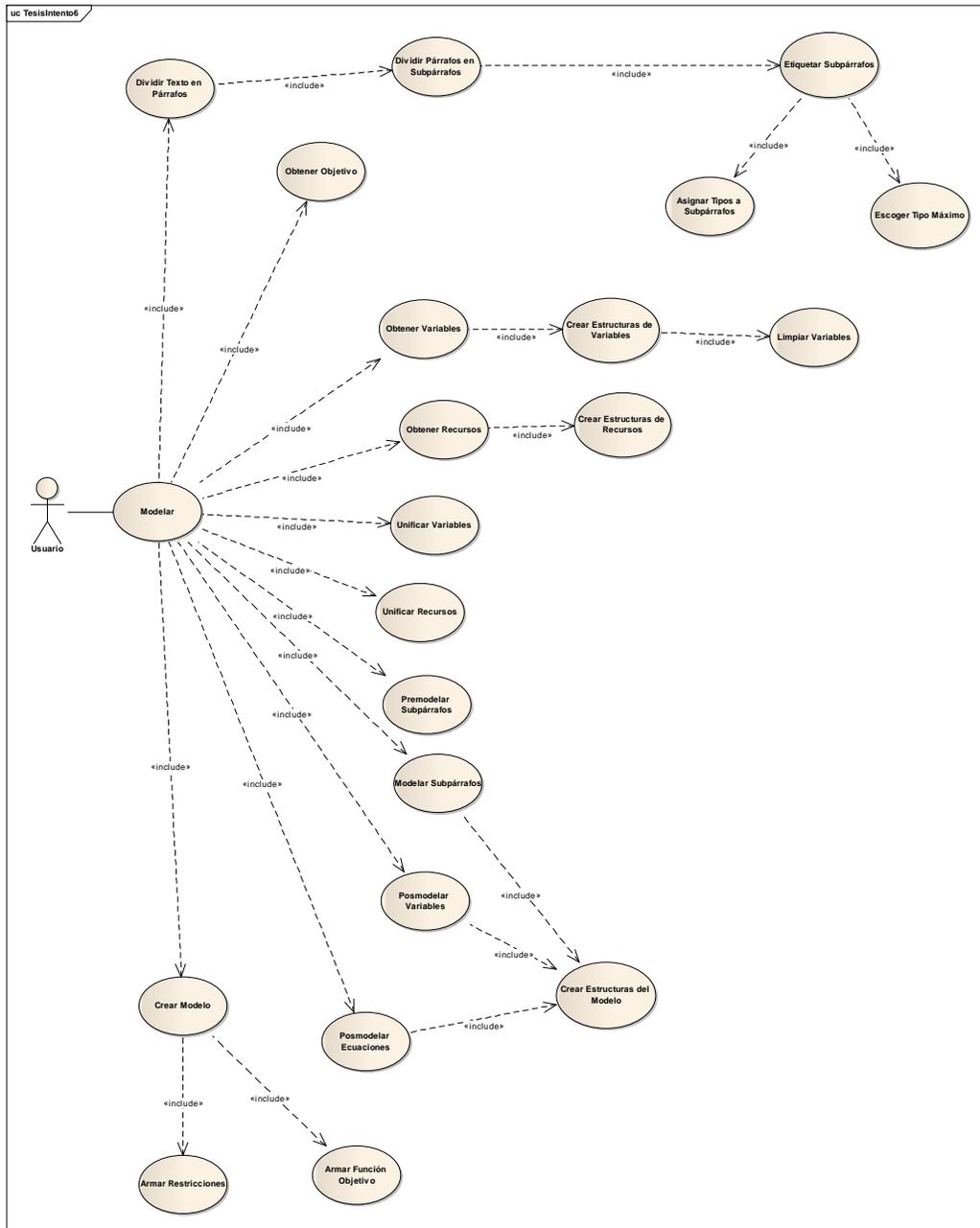


Diagrama específico: Detalladamente dichas acciones, implican los subprocesos presentados a continuación (Figura 45):

Figura 45 - Diagrama de Casos de Uso TesisIntento6



4.6.2 Diseño

Diagrama general de datos: En este intento, se logra el objetivo de encontrar correctamente los elementos del modelo, por lo que se deben implementar la fase de construcción del modelo y perfección de todo el proceso de modelado, con este fin, a nivel de base de datos se adicionan nuevas tablas como se muestra en el siguiente diagrama (Figura 47).

Figura 47 - Diagrama General de Datos TesisIntento6



Diagrama Conceptual:

Figura 48 - Diagrama Conceptual TesisIntento6

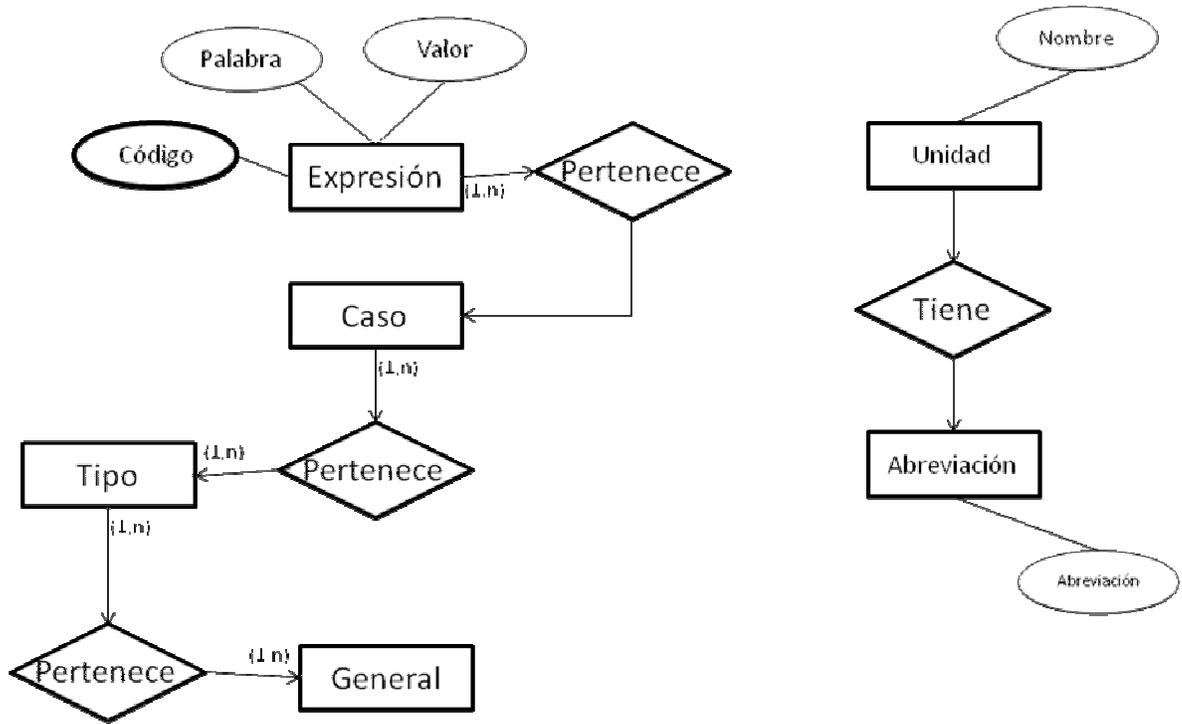


Diagrama de BD:

Estas nuevas tablas están compuestas y se añaden a la estructuras de la siguiente manera. Ver Figura 49:

Figura 49 - Diagrama de Base de Datos TesisIntento6

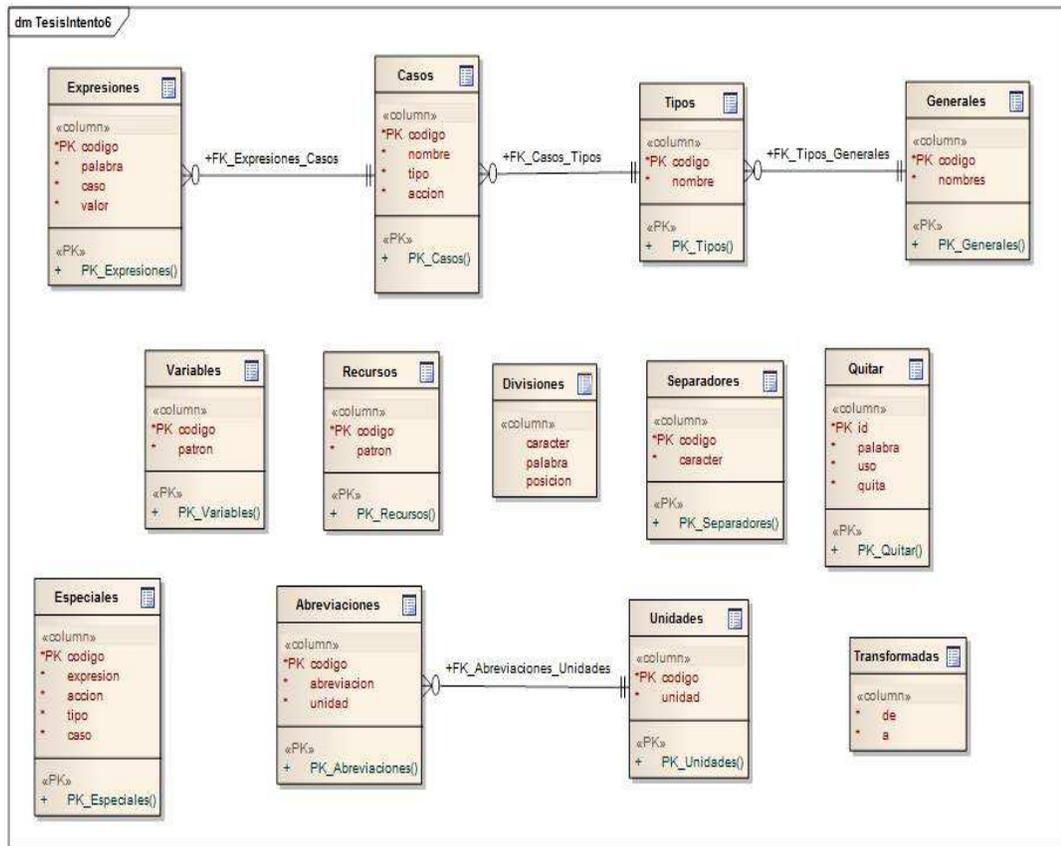


Diagrama de clases: En este intento se crea la estructura completa donde se almacena el modelo del ejercicio, además se divide el general del tipo, y se crea una clase para manejar varios de estos. Ver Figura 50:

Figura 50 - Diagrama de Clases TesisIntento6

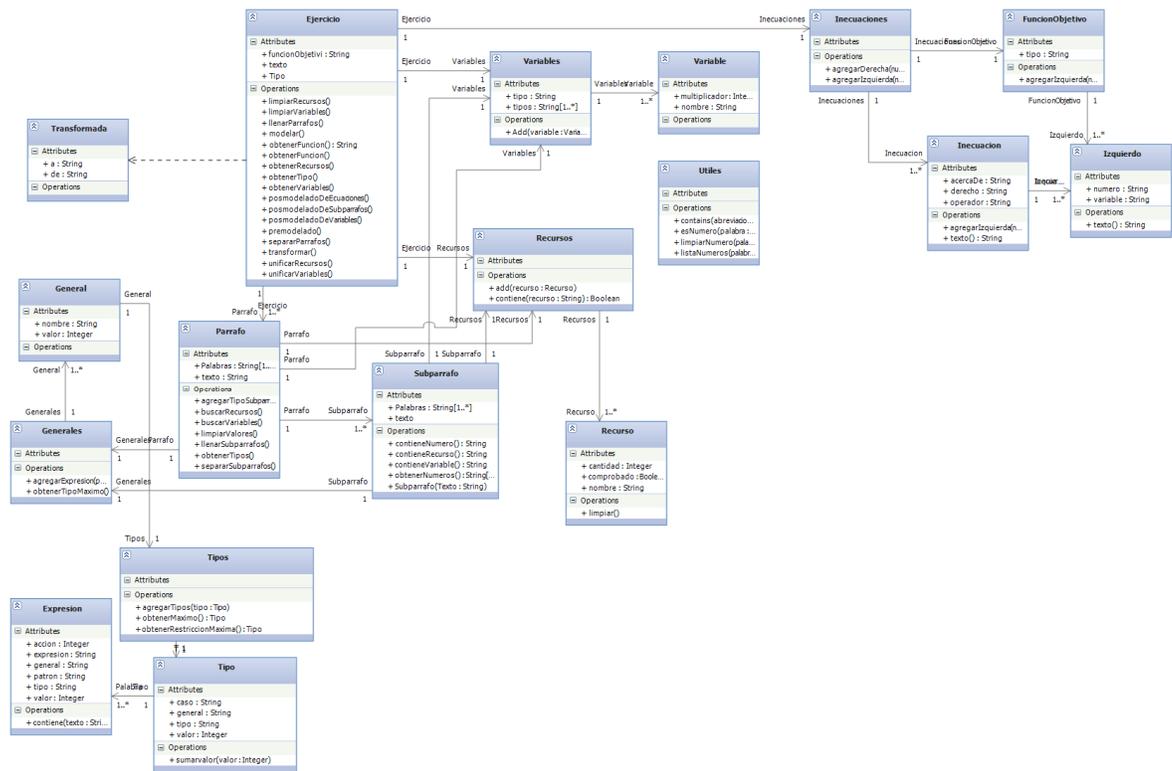
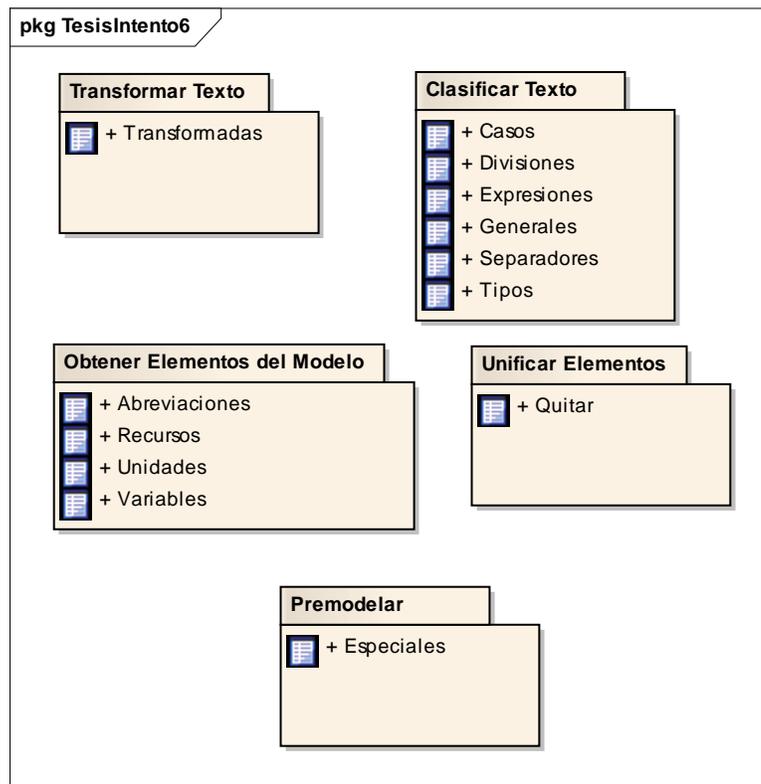


Diagrama de paquetes: En cuanto a funcionalidades, aparecen dos nuevas, que utilizan las nuevas tablas de la base de datos, como se muestra a continuación (Figura 51):

Figura 51 - Diagrama de Paquetes TesisIntento6



4.6.3 Pruebas de software: Las pruebas realizadas a Dionisio en cada iteración consistieron en alimentar al software con ejercicios, y los resultados fueron analizados para corregir las falencias encontradas, ya sea a nivel de diseño o de programación

Cada TesisIntento fue probado con veinte ejercicios escogidos por presentar variaciones en cuanto a objetivo, forma de presentar variables, recursos e inequaciones, estos ejercicios fueron previamente divididos, clasificados y modelados; para comparar los resultados obtenidos por Dionisio.

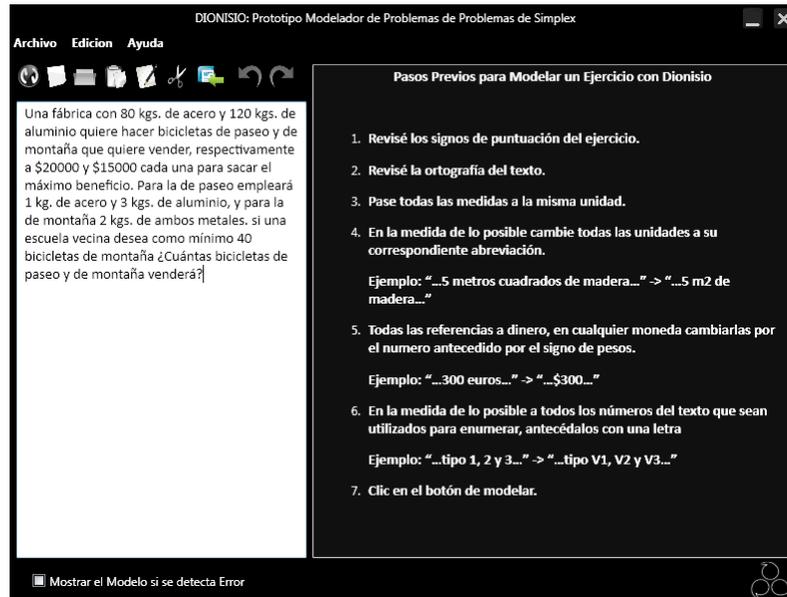
A continuación se indica una de estas pruebas realizada a la última iteración del software.

Ejercicio para la prueba:

Una fábrica con 80 kgs. de acero y 120 kgs. de aluminio quiere hacer bicicletas de paseo y de montaña que quiere vender, respectivamente a \$20000 y \$15000 cada una para sacar el máximo beneficio. Para la de paseo empleará 1 kg. de acero y 3 kgs. de aluminio, y para la de montaña 2 kgs. de ambos metales. si una escuela vecina desea como mínimo 40 bicicletas de montaña ¿Cuántas bicicletas de paseo y de montaña venderá?

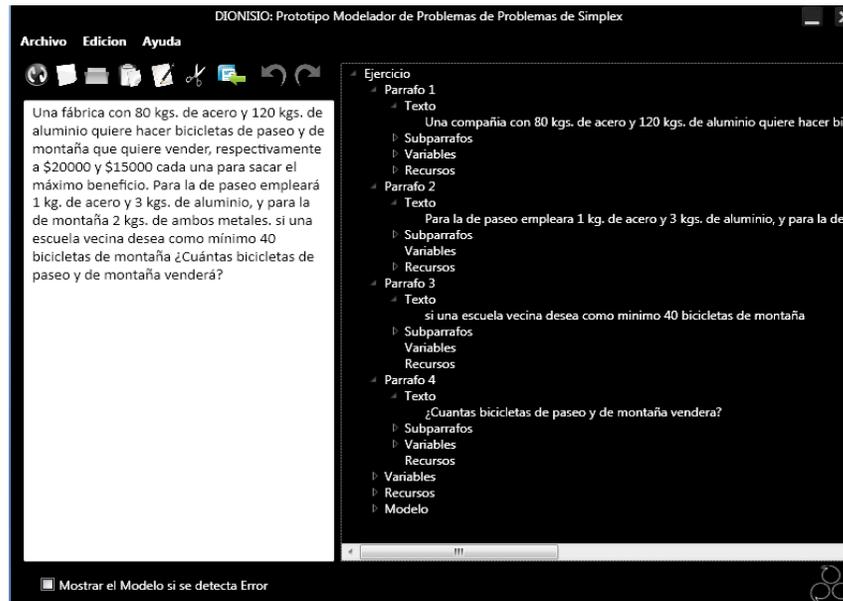
Como se muestra en la siguiente imagen (Figura 52), este ejercicio es introducido a Dionisio para ser modelado

Figura 52 - Prueba Ejercicio Escrito



Lo primero que se revisa en Dionisio es que la división del texto se realice coherentemente (Figura 53).

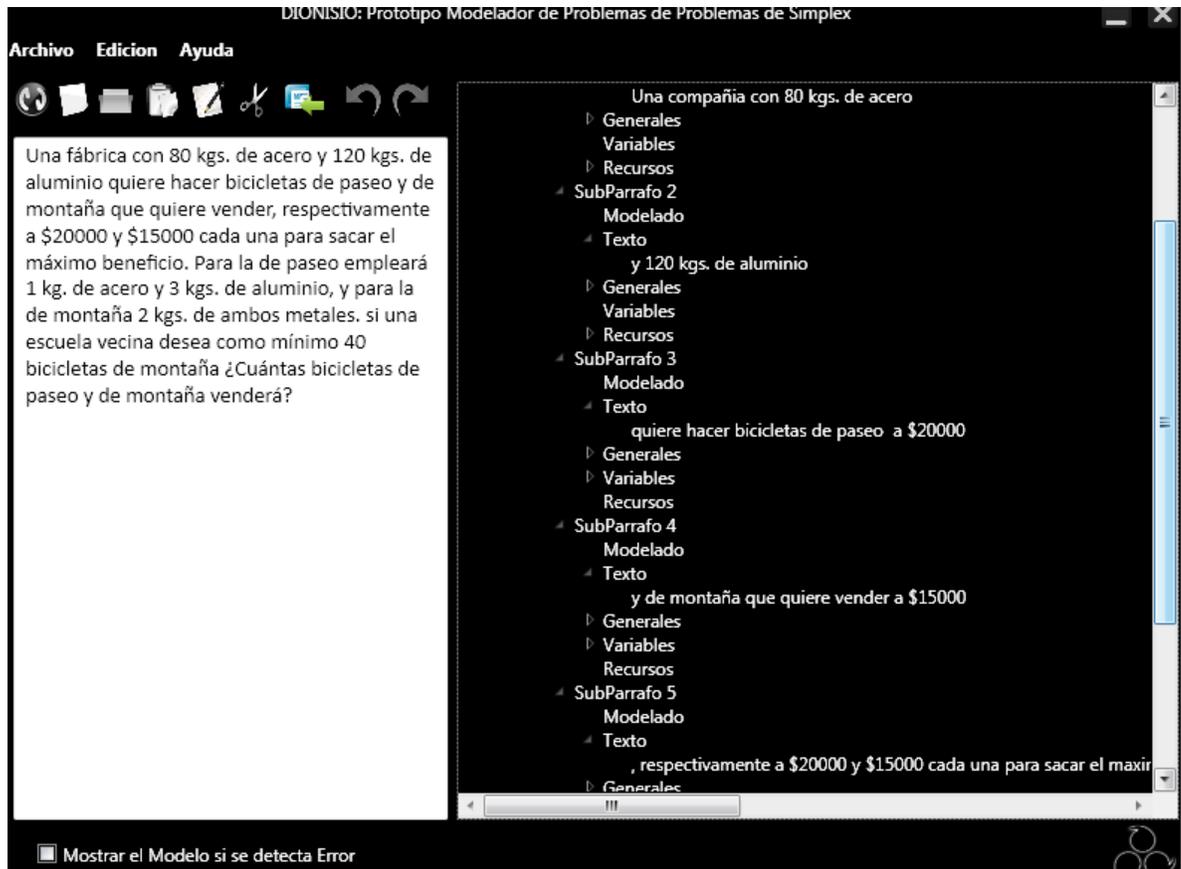
Figura 53 - Prueba División Texto



Como se puede observar en el pantallazo anterior (Figura 53), todos los párrafos y sus estructuras han sido divididos y construidas correctamente.

Paso siguiente, se revisa que todos los subpárrafos hayan sido divididos y armados correctamente.

Figura 54 - Prueba - Prueba División Párrafo

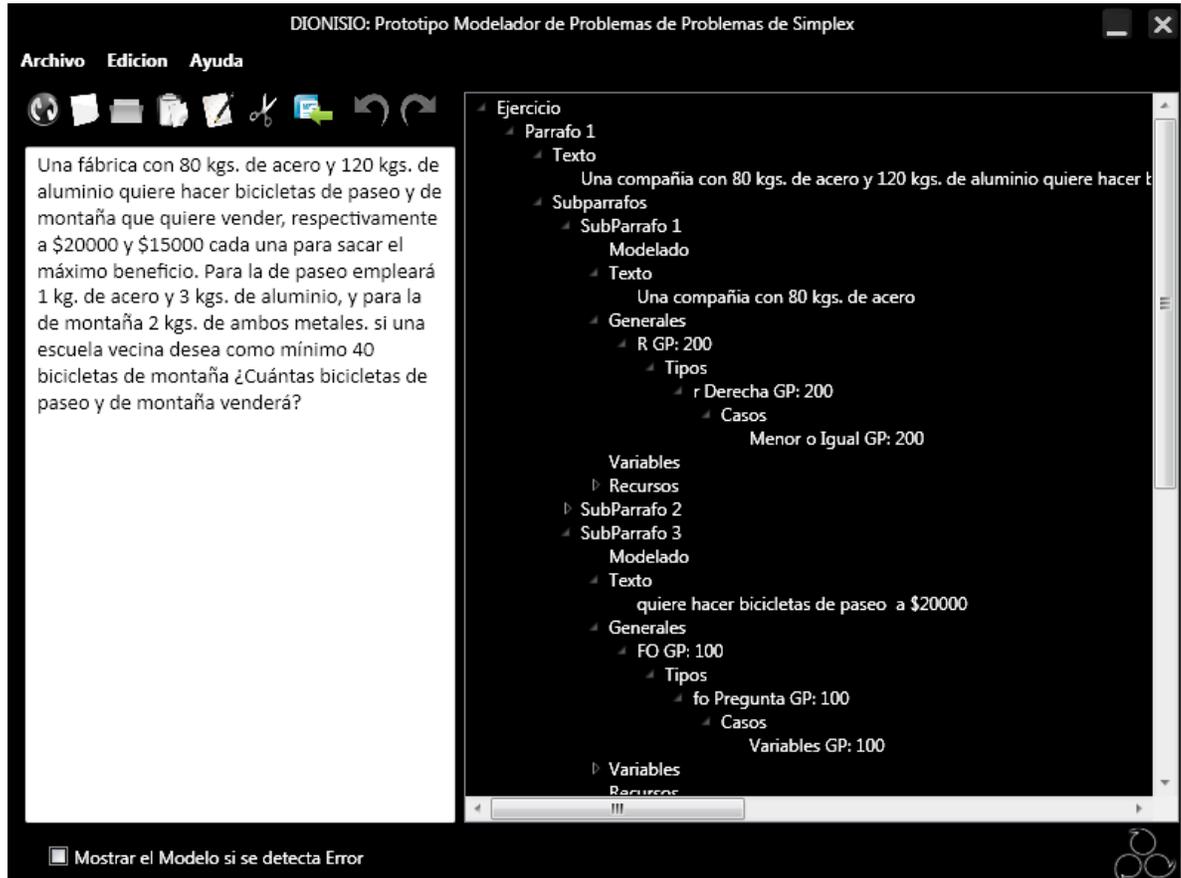


Como se puede ver (Figura 54), en este ejercicio hay dos casos especiales en la división de subpárrafos, el primero, cuando Dionisio encuentra divisores especiales, como en este caso lo es, "quiere hacer", se verifica que la expresión haya sido encontrada en el párrafo y que a partir de esta empiece un nuevo subpárrafo.

El segundo caso, es una excepción a la regla de división por la palabra "y". Se revisa que en el momento que encuentre una expresión como "respectivamente" no se divida por esta palabra.

Después de verificar la correcta división del texto, se prosigue a verificar la clasificación de los subpárrafos.

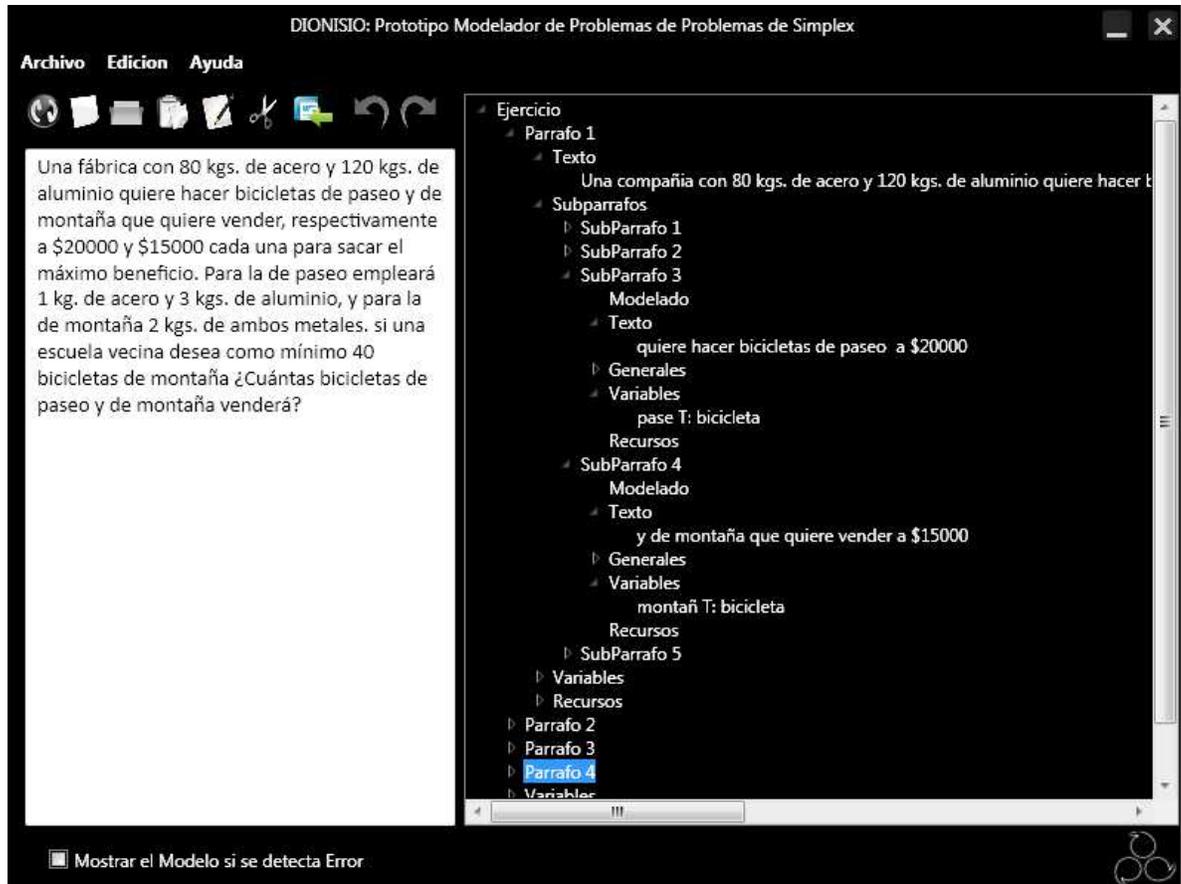
Figura 55 - Prueba Clasificación Subpárrafos



En esta parte es importante verificar que todas las expresiones sean identificadas correctamente en el párrafo, y que el grado de pertenencia de cada subpárrafo sea calculado correctamente. Con esta información se puede establecer que la clasificación se ha realizado con éxito, tal cual se muestra en la anterior imagen (Figura 55).

El siguiente proceso a verificar, es el de obtención de elementos del modelado para ello se empieza revisando las variables del ejercicio y su correcta obtención en los subpárrafos correctos.

Figura 56 - Prueba Obtención de Variables



A partir de la imagen anterior (Figura 56), se revisan los párrafos donde se enuncian las variables y se comparan con los patrones de obtención verificando cual se encuentra en dicho subpárrafo y se verifica que se agreguen a la estructura cada variable encontrada, con su respectivo tipo de variable si es que lo tiene.

El mismo proceso se repite con los recursos (Figura 57).

Figura 57 - Prueba Obtención de Recursos



Por último, con respecto a la obtención de elementos para el modelado, basándose en la siguiente imagen (Figura 58), se verifica que todos los recursos y variables sean unificados correctamente en todos los subpárrafos donde se los menciona y que además se guarden correctamente en la estructura general de Dionisio.

Figura 58 - Prueba Unificación



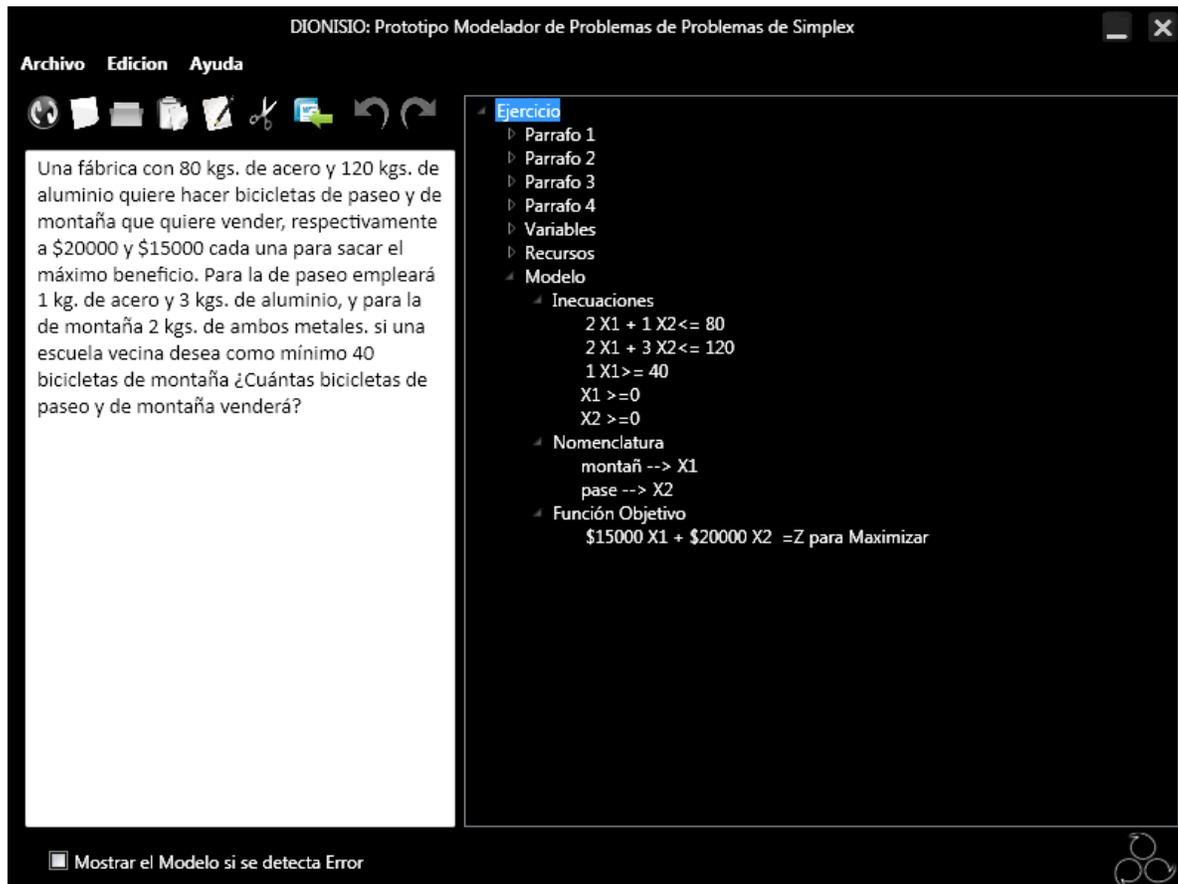
Ahora, solo queda verificar la construcción del modelo, para esto lo primero que se revisa es la estructura creada del modelo, listada abajo en la imagen (Figura 59), para así verificar parte por parte todas las restricciones creadas por Dionisio.

Figura 59 - Prueba Estructura del Modelo

Name	Value	Type
Inecuacioness.RestriccionesVarias	Count = 3	System.Collections.Generic.List<TesisFinal_f4.Inecuacion>
[0]	{TesisFinal_f4.Inecuacion}	TesisFinal_f4.Inecuacion
acercaDe	"acero"	string
AcercaDe	"acero"	string
Der	"80"	string
der	"80"	string
Izquierdos	Count = 2	System.Collections.Generic.List<TesisFinal_f4.Izquierdo>
[0]	{TesisFinal_f4.Izquierdo}	TesisFinal_f4.Izquierdo
Numero	"1"	string
numero	"1"	string
Texto	" + 1 X2 "	string
Variable	"X2"	string
variable	"X2"	string
[1]	{TesisFinal_f4.Izquierdo}	TesisFinal_f4.Izquierdo
Numero	"2"	string
numero	"2"	string
Texto	" + 2 X1 "	string
Variable	"X1"	string
variable	"X1"	string
Raw View		
Capacity	4	int
Count	2	int
Static members		
Non-Public members		
izquierdos	Count = 2	System.Collections.Generic.List<TesisFinal_f4.Izquierdo>
[0]	{TesisFinal_f4.Izquierdo}	TesisFinal_f4.Izquierdo
Numero	"1"	string
numero	"1"	string
Texto	" + 1 X2 "	string
Variable	"X2"	string
variable	"X2"	string
[1]	{TesisFinal_f4.Izquierdo}	TesisFinal_f4.Izquierdo
Numero	"2"	string
numero	"2"	string
Texto	" + 2 X1 "	string
Variable	"X1"	string
variable	"X1"	string
Raw View		
Capacity	4	int
Count	2	int
Static members		
Non-Public members		
Oper	"Menor o Igual"	string

Además, se realiza la comparación del modelo generado por Dionisio, presentado en la siguiente imagen (Figura 60), con el modelo real del ejercicio, para comprobar que no existen discrepancias.

Figura 60 - Prueba Modelo Final



4.6.4 Resultados obtenidos: Después de este intento Dionisio realiza sus procesos en once pasos, siguiendo el siguiente orden:

1. Obtención del objetivo.
2. División del texto.
3. Clasificación de las partes resultantes de la división, utilizando los patrones correspondientes al objetivo del problema.
4. Búsqueda de variables utilizando los patrones correspondientes al objetivo del problema, limitándose únicamente a las partes del texto en las cuales sean aplicables.
5. Limpieza y unificación de variables.
6. Búsqueda de recursos utilizando los patrones correspondientes al objetivo del problema, limitándose únicamente a las partes del texto en las cuales sean aplicables.
7. Limpieza y unificación de recursos.

8. Premodelado, que es el proceso mediante el cual Dionisio ubica la información necesaria para la creación del modelo en los lugares donde se la necesita.
9. Modelado, que es el proceso mediante el cual Dionisio construye el modelo llenando las diferentes estructuras creadas para el modelo, entre las que merecen especial atención aquellas que manejan las ecuaciones e inecuaciones que conforman un modelo; el llenado de estas estructuras se realiza mediante la aplicación de reglas de modelado.
10. Postmodelado, que es el proceso mediante el cual Dionisio modela información que se pudo pasar por alto en los anteriores pasos.
11. Presentación del modelo.

Se resalta que el cambio que destrabó la investigación fue el hecho de tener en cuenta información importante que se estaba pasando por alto en los anteriores intentos. Esto hizo que la obtención de los elementos del modelo se llevara a cabo con éxito, por lo que, a partir de este intento se pudo pasar a la siguiente fase consistente en la construcción del modelo. Estos cambios conllevaron obligadas modificaciones de algunos tramos del código y de la base de datos.

Este flujo final de procesos, se explica con más detalle en una sección posterior del documento que está dedicada a explicar los pasos que Dionisio lleva a cabo para lograr modelar los problemas.

Además, se obtuvo un resultado exitoso en las pruebas planteadas al software.

Conclusiones

Las conclusiones que se obtienen de este intento son ya, prácticamente, las mismas conclusiones del proyecto. Por ello se mencionan solamente las que se consideran más destacadas:

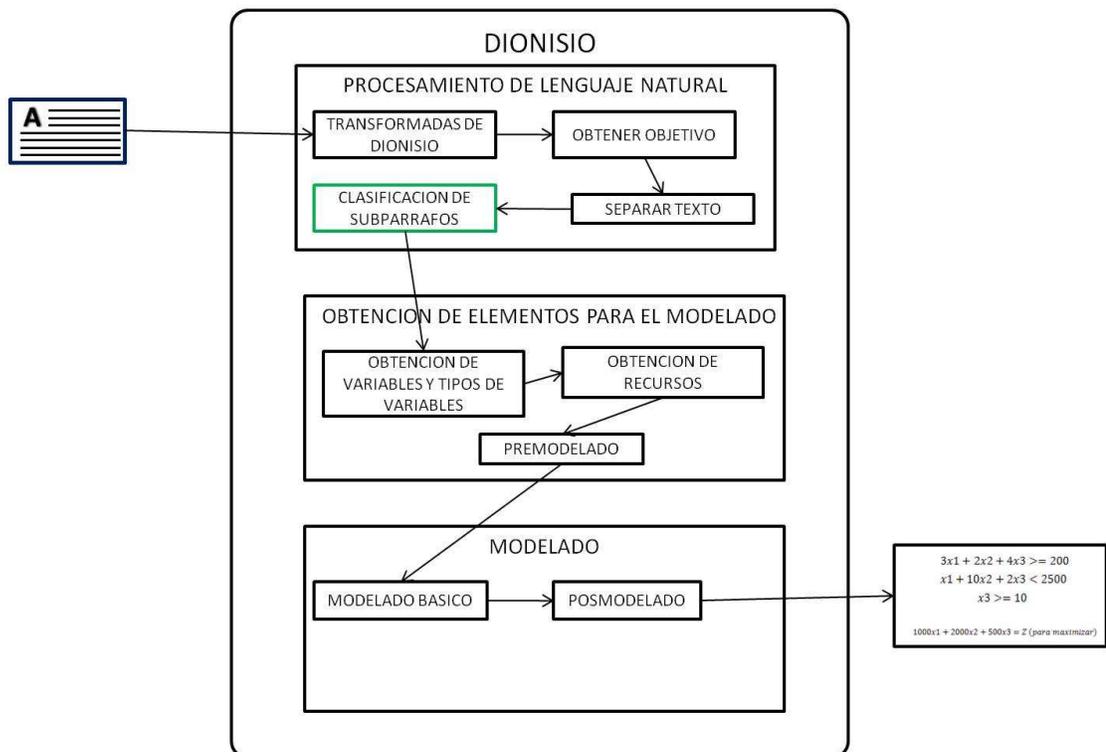
- Los patrones utilizados para clasificar y obtener los elementos del modelo deben variar según el objetivo del problema.
- Después de lograr obtener los elementos del modelo con éxito, la construcción del modelo es fácil a través de la aplicación de reglas básicas de modelado.
- Cada parte del modelo debe ser manejada a través de estructuras de información, lo que facilita el manejo y aprovechamiento de la información obtenida.
- Los procesos de premodelado y postmodelado son importantes para capturar información del modelo que se escapan a las reglas del modelo normal.

Trabajo posterior: Ahora, después de lograr los principales objetivos del proyecto se hace necesario trabajar en la interfaz de Dionisio para lograr una interfaz de usuario suficientemente amable. Entre tanto, es indispensable que la base de datos siga alimentándose y actualizándose para lograr que Dionisio sea capaz de modelar un mayor número de problemas.

5. DIONISIO

A través del siguiente diagrama (Figura 61), se describen los diferentes procesos y subprocesos que Dionisio lleva a cabo para modelar un problema de programación lineal.

Figura 61 - Diagrama de Entidades Nivel 1



Se explica ahora en qué consiste cada proceso y sus respectivos subprocesos.

5.1 PROCESAMIENTO DE LENGUAJE NATURAL

Es el proceso inicial. En esta etapa Dionisio toma como entrada el texto del problema de programación lineal, redactado en lengua castellana, exactamente como lo escribe el usuario. Inmediatamente procesa ese texto de tal manera que

Dionisio tenga los datos que necesita, en un formato apropiado, con el fin de convertirlos en información que incluya el objetivo del problema, las variables y los recursos.

Es la etapa más importante del proceso de Dionisio, ya que prepara la información con la que trabajan las siguientes etapas y debe, además, extraer correctamente los datos del problema. Es también la más difícil, porque la información con la que trabaja es el problema escrito, debiendo enfrentar las dificultades inherentes al manejo del lenguaje natural.

Para lograrlo, en esta fase del proceso se transforma, divide y clasifica el texto utilizando los diferentes subprocesos que a continuación se detallan.

5.1.1 Transformadas de Dionisio: Es la primera acción que Dionisio realiza y consiste en la realización de pequeñas transformaciones o traducciones del texto original del problema, para hacer que palabras o expresiones que puedan confundir a Dionisio sean cambiadas por palabras y expresiones sinónimas como se indica en el siguiente ejemplo:

En la frase:

“Una **fábrica** ofrece al mercado cuatro productos...”

La palabra “fábrica” causa ambigüedad ya que Dionisio no tiene como identificar si se refiere a “fábrica” como sujeto de la oración o a “fabrica” como verbo de la misma, debido a que, llegado el momento de comparar esta palabra con los patrones que utiliza Dionisio, éste quita las tildes para facilitar el proceso. Por ello se aplica una transformada en donde la palabra “fábrica” sea cambiada por la palabra “compañía” por lo que la frase resultante con la que trabaja Dionisio es:

“Una **compañía** ofrece al mercado cuatro productos...”

Por medio de esta transformada se soluciona un problema de ambigüedad en el texto del problema. Además de esto, las transformadas ayudan a Dionisio a:

- Reescribir subpárrafos con ecuaciones implícitas para que sean procesados más fácilmente.

- Separar subpárrafos con mucha información, que no podrían separarse de otra manera.
- Organizar el texto dentro de los subpárrafos para que concuerden con los patrones.

5.1.2 Obtener objetivo: Los modelos matemáticos de programación lineal cuentan con un objetivo (Maximizar o Minimizar) que está inmerso en el texto del problema, por lo que hay que extraerlo a través de la identificación de patrones presentes en el texto, que para Dionisio indican objetivos. Esta etapa es importante ya que, dependiendo del objetivo del problema, Dionisio determina los patrones que utilizará y en qué forma lo hará para llevar a cabo los procesos futuros de clasificación de texto y obtención de elementos para el modelado.

5.1.3 Separar texto: Basándose en el principio del análisis sistémico, Dionisio empieza el proceso de modelado a través de la división del texto del problema para reducirlo a una unidad de información que pueda ser clasificada y de la que se pueda obtener datos puntuales del problema. Esta división de texto se hace en dos niveles, como se muestra a continuación.

Separar párrafos: El primer nivel de división de texto que hace Dionisio. Aquí se arman párrafos dividiendo el texto del problema por cada punto (.) presente en el texto.

Separar subpárrafos: El segundo nivel de división de texto que hace Dionisio. Aquí se arman subpárrafos separando los párrafos resultantes del paso anterior, a través de coma (,), punto y coma (;), dos puntos (:), símbolos de interrogación (¿,?) y por expresiones que implican cambio de idea dentro de los textos de problemas de programación lineal.

Los subpárrafos que arma Dionisio no son sólo partes del texto del problema. Por conveniencia, cada subpárrafo es una estructura de información donde se almacenan las variables y los recursos que contiene el subpárrafo, las palabras que lo componen, las posibles clasificaciones y una clasificación definitiva, además de un estado, para determinar más adelante si ya ha sido modelado.

5.1.4 Clasificación de subpárrafos: Es el proceso más importante que se realiza, ya que la clasificación determina el lugar del texto donde Dionisio debe buscar para obtener cada elemento del modelo matemático. Para realizar la clasificación, Dionisio utiliza lógica difusa ya que con esta técnica se puede tratar con relativa facilidad la ambigüedad homónima presente en el texto en cuanto a la semántica del subpárrafo.

Con el texto dividido Dionisio clasifica cada subpárrafo de la siguiente manera:

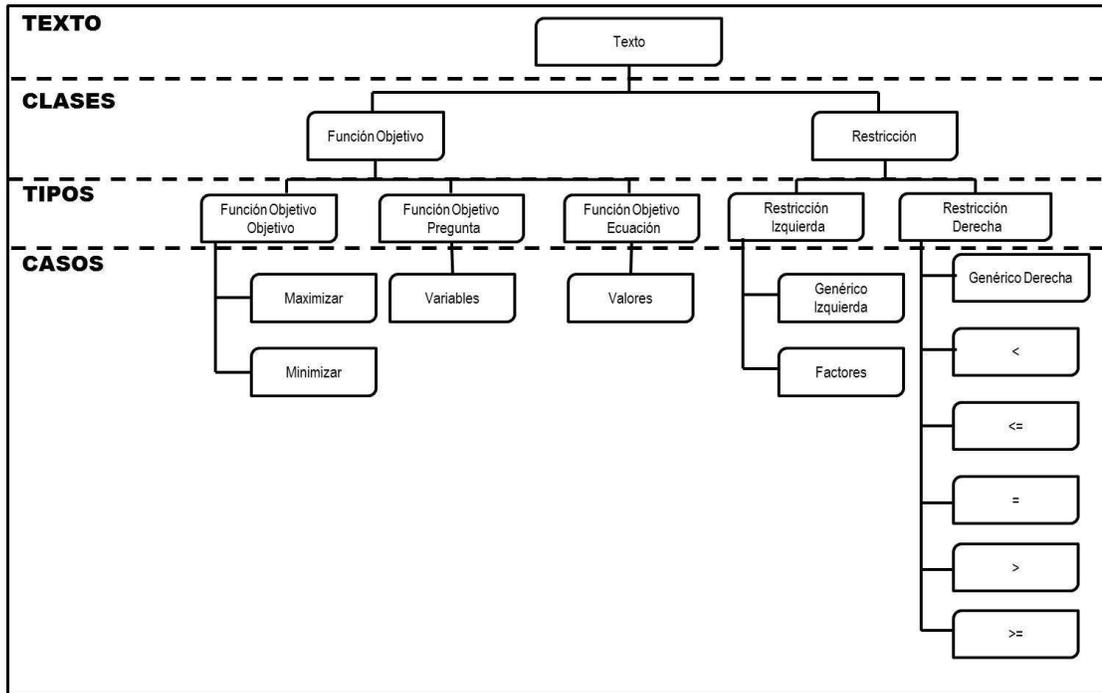
Búsqueda de expresiones: En este primer paso de la clasificación Dionisio busca patrones que se encuentren dentro de dicho subpárrafo y los guarda en una lista con sus respectivos pesos.

Asignación de pesos a subpárrafos: Como un subpárrafo puede contener diferentes patrones, Dionisio los agrupa según su tipo y después suma sus pesos.

Etiquetado de subpárrafos: Después de agrupar y sumar los pesos, Dionisio escoge la clasificación más adecuada para el subpárrafo y se la asigna. En caso de que el proceso de asignación de pesos no de un resultado concluyente acerca del tipo de subpárrafo, la clasificación queda desierta por lo que posteriormente se aplicará al párrafo las reglas de herencia y contexto.

El proceso de etiquetado se realiza basándose en la siguiente jerarquía (Figura 62):

Figura 62 - Jerarquía de Clasificación



Esta jerarquía fue construida para este trabajo de investigación, identificando cada parte y subparte de un modelo matemático de programación lineal como se explica a continuación, a través de un ejemplo.

En el siguiente modelo,

$$200x_1 + 240x_2 = Z \text{ Para Minimizar}$$

$$x_1 \leq 10$$

$$x_2 \leq 5$$

$$40x_1 + 30000x_2 = 400000$$

$$2x_1 - x_2 \geq 15$$

, en primera instancia se puede identificar dos partes:

- Función Objetivo:

$$200x_1 + 240x_2 = Z \text{ Para Minimizar}$$

En la jerarquía propuesta, esta ecuación corresponde a la clase Función Objetivo.

- Restricciones:

$$x_1 \leq 10$$

$$x_2 \leq 5$$

$$40x_1 + 30000x_2 = 400000$$

$$2x_1 - x_2 \geq 15$$

$$x_1, x_2 \geq 0$$

En la jerarquía cada inecuación corresponde a la clase Restricción.

Ahora, para seguir detallando la jerarquía propuesta, se atiende en primer lugar de la Función Objetivo, en la cual se encuentran los siguientes componentes:

$$200x_1 + 240x_2 = Z \text{ Para Minimizar}$$

- Objetivo de función objetivo: En este caso Minimizar. Para esta parte del modelo se creó el tipo Función Objetivo Objetivo, el cual puede ser de dos casos Maximizar o Minimizar. A este tipo pertenecerán los patrones que indique función objetivo y su caso se determinará obviamente por el objetivo que representen.
- Variables de función objetivo: En este caso x_1, x_2 , por esta parte se creó el tipo Función Objetivo Pregunta, que solo puede ser de caso Variables, a este tipo y caso pertenecen los patrones que sirven para identificar variables.

Valores de función objetivo: En este caso 200 y 240. Para esta parte se creó el tipo Función Objetivo Ecuación, que sólo puede ser de caso Valores. A este tipo y caso pertenecen los patrones que sirven para identificar números dentro del texto.

Esta es la parte de la jerarquía que corresponde a la Función Objetivo del Problema. Ahora se explica la jerarquía que nace de la Restricción. Para ello se tomará una restricción del modelo ejemplo.

En la restricción,

$$40x_1 + 30000x_2 = 400000$$

se puede encontrar las siguientes partes:

- Parte izquierda: $200x_1 + 240x_2$ Para esta parte se crea lo que, en este trabajo se denominó Restricción Izquierda, que puede tener los siguientes casos:
 - Genérico izquierda: pertenecen los patrones que indican que una variable consume o requiere un recurso.
 - Factores: pertenecen los patrones que sirven para identificar los números dentro del texto. En este caso los factores son 40 y 30000.
- Parte derecha: $= 400000$ Para esta parte se crea lo que se denomina Restricción Derecha, que puede tener los siguientes casos:
 - Genérico derecha: pertenecen los patrones que indican un límite a los recursos.
 - $<$, \leq , $=$, $>$, \geq : pertenecen los patrones que indican el signo de las ecuaciones dentro del texto.

Después de determinar esta jerarquía se la transfiere a Dionisio a través de su diseño de Base de Datos.

Ahora se presenta un ejemplo de cómo Dionisio clasifica un subpárrafo.

En el subpárrafo, “¿Cuántas unidades de cada producto son necesarias a fin de maximizar las utilidades?”, Dionisio encuentra dos posibles clasificaciones ya que la expresión “...maximizar las utilidades...” indica que éste puede ser un subpárrafo de Función Objetivo Objetivo, pero además, la expresión “...son necesarias...” indica que éste mismo puede ser un subpárrafo de Restricción Izquierda. En este caso Dionisio utiliza lógica difusa para encontrar la mejor opción entre estas dos posibilidades.

Reglas de herencia y contexto: Como se mencionó, existe la posibilidad de que un subpárrafo quede sin clasificar. Cuando esto ocurre, Dionisio utiliza las siguientes reglas para asignarle una clasificación:

- Si el subpárrafo contiene una expresión que indica continuidad de la idea del subpárrafo anterior se le hereda la clasificación de este subpárrafo.
- En caso de no existir la expresión mencionada en la regla anterior, el subpárrafo es clasificado según su contexto, asignándole un promedio de las clasificaciones de todos los subpárrafos pertenecientes al párrafo del subpárrafo sin clasificar.

5.2 OBTENCIÓN DE ELEMENTOS PARA EL MODELADO

Al llegar a esta fase, Dionisio ya ha identificado el objetivo del problema y ha clasificado cada parte del texto. El paso siguiente, de acuerdo con esa información, consiste en extraer los datos que le permitan a Dionisio construir el modelo. Para ello, Dionisio se centra en obtener las variables y los recursos que vienen en el enunciado del problema.

5.2.1 Obtención de variables y tipos de variables

Por medio de la búsqueda de patrones en el texto se identifican las posibles variables del problema y sus respectivos tipos.

Detección de patrones a utilizar: Los patrones que Dionisio utiliza para dicho reconocimiento dependen en algunos casos del objetivo del problema. Por ello, los patrones están clasificados en tres categorías:

- Patrones de Variables de Maximización: Son los patrones que se activan en caso de que el objetivo del problema sea maximizar.
- Patrones de Variables de Minimización: Son los patrones que se activan en caso de que el objetivo del problema sea minimizar.
- Patrones de Variables de Mixtos: Estos patrones se activan sin importar el objetivo del problema.

Además de que la aplicación de un patrón depende del objetivo del problema, también depende de la clasificación del texto. Por eso, los patrones de variables vuelven a estar categorizados de la siguiente manera:

- Patrones de Variables de FO: se aplican en los subpárrafos de tipo FO.
- Patrones de Variables de Restricciones: se aplican en los subpárrafos de tipo Restricción.
- Patrones de Variables Independientes: se aplican en todos los subpárrafos sin importar su clasificación.

A partir de las anteriores categorías y dependiendo del tipo de problema, Dionisio escoge los patrones pertinentes para ser aplicados.

Búsqueda de patrones y depuración de variables: Después de determinar los patrones que debe aplicar, Dionisio realiza la búsqueda, la cual da como resultado un listado de variables y un listado de tipos iniciales.

Estos listados están compuestos por expresiones extraídas directamente del texto, tal como fueron escritas y ello implica que dichas expresiones, en muchas ocasiones están escritas en un determinado tiempo verbal. Por esta razón se hace necesario que Dionisio lleve a cabo un proceso de depuración, eliminando prefijos y sufijos para obtener la raíz de la expresión, al tiempo que elimina la información irrelevante que acompaña a la posible variable o tipo. Esa información puede constar de números, artículos, pronombres, adverbios y otras partes invariables de la oración. Esta eliminación de información irrelevante se lleva a cabo a través de una tabla de la base de datos que almacena las palabras o expresiones que se consideraron basura porque siempre aparecen pero carecen de significancia.

Unificación de variables: Con la realización de la depuración de las variables y sus tipos, Dionisio da por terminado el proceso de obtención de estos elementos del modelo. Ahora Dionisio procede a asignar cada variable y tipo encontrado a cada subpárrafo siempre y cuando dicha variable o tipo esté presente en el texto del subpárrafo o el subpárrafo haga referencia a él. A este proceso se le denomina unificación de tipos y variables.

Dado que, para realizar esta acción no se igualan las variables con las palabras del texto, sino que se mira si están contenidas o no, se hizo necesaria la creación de una tabla que se denominó Inconfundibles, siendo éstas, palabras o expresiones que Dionisio no puede confundir con variables o recursos al momento de unificarlos. Entre éstos se encuentran, los artículos de las oraciones.

5.2.2 Obtención de recursos: Con algunas variaciones, la obtención de recursos que lleva a cabo Dionisio, sigue los mismos pasos empleados para la obtención de variables.

Se destaca que los patrones utilizados en la obtención de variables, son diferentes de los utilizados en la obtención de recursos y que algunos de los patrones de obtención de recursos tienen en cuenta las variables obtenidas en el paso anterior. Por ello, la obtención de recursos se lleva a cabo sólo cuando la obtención de variables haya terminado.

Detección de patrones a utilizar: Al igual que en el caso de las variables, los patrones que Dionisio utiliza para este reconocimiento dependen en algunos casos del objetivo del problema, justificando que los patrones estén clasificados en tres categorías:

- Patrones de recursos de maximización: se activan en caso de que el objetivo del problema sea maximizar.
- Patrones de recursos de minimización: se activan en caso de que el objetivo del problema sea minimizar.
- Patrones de recursos de mixtos: se activan sin importar el objetivo del problema.

Además de que la aplicación de un patrón depende del objetivo del problema, también depende de la clasificación del texto. Como consecuencia, los patrones de variables vuelven a estar categorizados de la siguiente manera:

- Patrones de variables de FO: se aplican en los subpárrafos de tipo FO.
- Patrones de variables de restricciones: se aplican en los subpárrafos de tipo Restricción.
- Patrones de variables independientes: se aplican en todos los subpárrafos sin importar su clasificación.

A partir de las anteriores categorías y dependiendo del tipo de problema, Dionisio escoge los patrones pertinentes para ser aplicados.

Búsqueda de Patrones y Depuración de Recursos: Después de determinar qué patrones debe aplicar, Dionisio realiza la búsqueda de ellos, y obtiene como resultado un listado de recursos iniciales. Este listado, al igual que el de las variables y el de los tipos, está compuesto por expresiones extraídas directamente del texto, tal como fueron escritas, por lo que se lleva a cabo el mismo proceso de depuración descrito anteriormente.

Unificación de Recursos: Al finalizar la depuración del listado de recursos, Dionisio ya cuenta con este elemento del modelo y, al igual que con las variables, los recursos deben ser unificados para tenerlos en los subpárrafos, donde en el futuro se necesiten. Como se hizo con las variables, también aquí se aplican los Inconfundibles.

5.2.3 Premodelado: En esta etapa Dionisio se encarga de preparar la estructura de datos y el texto de los subpárrafos añadiendo información que para un ser humano sería tácita pero que para Dionisio hasta este momento no había sido clara, como el uso de las expresiones *respectivamente, a ambos, de igual forma, cada una, etc.*

Este proceso permitirá a Dionisio modelar problemas donde cierta información del modelo aparezca implícita gracias a la utilización de lo que se denominó expresiones especiales. Cuando Dionisio encuentra una de ellas, agrega la información pertinente.

5.3 MODELADO

Con el procesamiento del texto llevado a cabo por Dionisio en las etapas anteriores, éste ya cuenta con la información necesaria para armar el modelo matemático del problema de programación lineal. Esta etapa se divide en unas subetapas que se describen a continuación.

5.3.1 Modelado básico: En esta etapa, Dionisio imita a un ser humano en la formulación del modelo, basándose en las siguientes reglas:

- Si en un subpárrafo de clase Función Objetivo se encuentra un número y una variable, ese número es el que determina el aporte de la variable a la función objetivo.
- Si en un subpárrafo de clase Restricción y tipo Restricción Izquierda, se encuentra un número, una variable y un recurso, este número es la cantidad de recurso que consume la variable.
- Si en un subpárrafo de clase Restricción y tipo Restricción Derecha se encuentra un número y un recurso, este número es la cantidad del recurso disponible. Por su parte la clasificación del subpárrafo determina el signo de la restricción dependiendo del caso ($<$, \leq , $=$, \geq , $>$).
- Si en un subpárrafo de clase Restricción y tipo Restricción Derecha se encuentra un número y una variable, este número es un límite de la variable, por su parte la clasificación del subpárrafo determina el signo de la restricción dependiendo del caso ($<$, \leq , $=$, \geq , $>$), cuando aplique esta regla Dionisio debe armar la parte izquierda de la restricción con un coeficiente de 1 para la variable encontrada.

De acuerdo con estas reglas, Dionisio arma una estructura de información para cada elemento del modelo con sus respectivos datos y cambia el estado del subpárrafo de donde extrajo la información pasándolo a subpárrafo modelado. Después de este paso, Dionisio ya cuenta con un modelo inicial del problema. Sin embargo, todavía existen elementos del modelo que escapan de las reglas anteriormente mencionadas por lo que es necesario someter los subpárrafos no modelados a un tratamiento especial de posmodelado.

5.3.2 Posmodelado

En este proceso, Dionisio intenta modelar los subpárrafos que no fueron modelados anteriormente, realizando dos subprocesos.

Posmodelado de variables: En este posmodelado, Dionisio busca subpárrafos no modelados y aplica las siguientes reglas:

- Si un subpárrafo no modelado es de clase Restricción y tipo Restricción Derecha y en él se encuentra un número y un tipo de variable, este número restringe a la sumatoria de todas las variables que van en la parte izquierda con coeficiente 1.

- Si un subpárrafo no modelado es de clase Restricción y tipo Restricción Derecha y en él se encuentra un número y una abreviación, este número restringe a la sumatoria de todas las variables que tengan esa abreviación. Estas variables van en la parte izquierda con coeficiente 1.

Dionisio nuevamente arma estructuras de restricciones según lo que identifique.

Posmodelado de ecuaciones: En esta fase Dionisio recorre las restricciones armadas hasta el momento y dependiendo de la estructura actúa así:

- Si la restricción no tiene operador en su estructura la elimina.
- Si la restricción no tiene en su estructura parte izquierda y si la parte derecha limita un tipo de variable, arma la estructura izquierda con las variables pertenecientes a ese tipo con coeficiente 1.
- Si después de esto persisten restricciones sin parte izquierda y, si la parte derecha limita una variable, arma la estructura izquierda con la variable con coeficiente 1.

5.4 PRESENTACIÓN DEL MODELO

La presentación del modelo resultante al usuario se realiza en un tree-view, donde se lista el texto dividido en estructuras de párrafos y subpárrafos con su respectiva clasificación y elementos para que sirva como guía para el usuario acerca de la forma en que se lleva a cabo el proceso de modelado. También se presentan los recursos y las variables con su respectiva nomenclatura. Por último, se listan las estructuras correspondientes a la función objetivo y las restricciones del problema con sus respectivas partes, ya como el modelo final abstraído del problema.

Si el proceso de modelado realizado por Dionisio falla en alguna de las etapas descritas anteriormente, se informara al usuario del posible error y una posible solución así:

- Cuando Dionisio no Determine el Objetivo del Ejercicio: Se informa al usuario que es posible que el texto digitado no corresponda a un ejercicio de programación lineal, ya que el objetivo es un elemento que distingue estos problemas. Sin embargo, cómo es posible que el objetivo del problema este contenido en un patrón que no sea reconocido por Dionisio, se lista algunos de los patrones más generales que reconoce Dionisio, en cuanto al objetivo, con los cuales el usuario puede formatear el texto del problema de tal forma que Dionisio reconozca el patrón y pueda obtener el objetivo.

- Cuando Dionisio no Determine las Variables del Ejercicio: Se informa al usuario que no se pudieron encontrar estos elementos debido a que no están en el texto o a que Dionisio no posee los patrones para reconocerlas, por esta razón se listan los patrones de reconocimiento más generales con ejemplos explicativos para que el usuario pueda formatear el texto de tal forma que Dionisio pueda determinar las variables del problema.
- Cuando Dionisio no Puede Construir las Restricciones: Se informa al usuario que no se pudieron construir las restricciones del problema se presentan las posibles causas y posibles soluciones con ejemplos explicativos para que el usuario pueda formatear el texto de tal forma que Dionisio pueda determinar los recursos del problema.

Si se presenta alguno de los anteriores errores, el usuario tiene la opción de ver el modelo generado aunque contenga errores, esto con el fin de que compare su clasificación con la llevada a cabo por Dionisio.

6. TRABAJOS FUTUROS

La creación de Dionisio como prototipo modelador de problemas de simplex, podría no tener fin, porque siempre será susceptible de mejoramiento. Por otra parte, Dionisio puede servir como base para crear aplicativos similares.

Nos permitimos sugerir algunas mejoras y algunos aplicativos afines.

- Actualización permanente de la base de datos con el fin de que Dionisio pueda modelar más ejercicios.
- Creación de un módulo de aprendizaje a través del cual Dionisio pueda expandir su base de datos cuando se enfrente a un problema nuevo.
- Con el fin de completar la temática de programación lineal, añadir a Dionisio los módulos que le permitan modelar problemas de asignación y transportes.
- Crear nuevos prototipos que modelen texto de problemas diferentes a los de programación lineal, en otros campos de la ciencia como física, química, matemática, algebra, etc. Allí, el procedimiento de construcción es similar contando con un léxico diferente.
- Dotar a Dionisio y a sus similares de otras formas de adquirir la información que sean diferentes al texto. Se sugiere en principio una cámara que le permita literalmente leer el texto.
- En un futuro, cuando la tecnología lo permita, crear una herramienta que no sólo modele texto sino que también modele realidades. En este caso el futuro Dionisio no recibirá el texto del problema que se desea modelar sino que lo producirá a partir de su percepción del ambiente que lo rodea.

7. CONCLUSIONES

Como primera medida fue necesario analizar los modelos matemáticos de programación lineal para identificar su estructura, sus partes y los elementos que lo componen, de este análisis se concluyó:

- Dentro de los modelos matemáticos de programación lineal hay cuatro tipos de estructuras que se deben armar a la hora de construir el modelo, la función objetivo, las restricciones de recursos, las restricciones de variables y las restricciones de no negatividad.
- La función objetivo es una ecuación que siempre debe estar presente en el modelo matemático y para este estudio debe ser única para cada problema.
- Los dos primeros tipos de restricciones pueden ser ecuaciones o inecuaciones su aparición en el modelo varía dependiendo del problema.
- Las restricciones de recursos están conformadas por variables que consumen al recurso y por un valor que lo limita.
- Las estructuras mencionadas anteriormente estas conformadas por los siguientes elementos:
 - Objetivo: Representa la finalidad del problema, puede ser Maximizar o Minimizar, siempre debe estar presente en los problemas de programación lineal, hace parte de la función objetivo.
 - Variables: Son las incógnitas del modelo, o sea los valores que se quieren determinar según el objetivo, siempre deben estar presentes en los problemas. Son los principales elementos del modelo ya que aparecen en la función objetivo, en las restricciones de recursos y obviamente en las restricciones de variables, en algunos problemas estas variables tienen tipos que las agrupan a todas a través de los cuales se arman las restricciones de variables.
 - Recursos: Son los elementos a partir de los cuales se establecen las restricciones de recursos, siempre que aparezcan deben estar limitados por un símbolo que indique igualdad o desigualdad, debe existir al menos una restricción por cada recurso.

- Existen símbolos que al unirse con los elementos mencionados anteriormente complementan un modelo matemático entre ellos están:
 - Números: Se pueden encontrar tres clases de números, los que acompañan a las variables en la función objetivo señalando el aporte de la variable al objetivo, los que acompañan a las variables en las restricciones señalando la cantidad de recurso que consume cada variable y los que limitan a los recursos en las restricciones.
 - Signos: Entre ellos están los signos matemáticos de suma y resta y los que definen el límite del recurso o la(s) variable(s) en las ecuaciones e inecuaciones, entre ellos se encuentran $<$, \leq , $=$, \geq , $>$.
- Una restricción de recurso limita a un recurso mientras un recurso puede ser limitado por más de una restricción de recurso.
- Las restricciones de variables están conformadas por las relaciones entre las variables y un límite para dicha relación.
- Una restricción de variables puede limitar a una o más variables, igualmente las variables pueden ser limitadas por una o más restricciones de variables.

Estos elementos y estructuras son extraídos del texto por el ser humano a la hora de modelar un problema de programación lineal, a partir de esto se desarrolló un prototipo de modelador de problemas de simplex denominado Dionisio, el cual modela problemas de programación lineal escritos en lenguaje natural tal como lo haría un ser humano.

Para construir Dionisio fue necesario analizar el proceso de modelado que realiza un ser humano, en este análisis se encontró que aunque el ser humano no se da cuenta se llevan a cabo muchos procesos en su mente antes de obtener el modelo matemático final, a continuación se presentan las conclusiones de este análisis:

- La primera etapa que el ser humano lleva a cabo inconscientemente y que no siempre se hace bien, es la división y clasificación del texto, en el transcurso de la primera lectura del problema el ser humano identifica que parte del texto contiene los datos que forman el modelo matemático, esta clasificación se lleva a cabo relacionando las partes del texto con las estructuras y elementos del modelo matemático descritas anteriormente.

- La abstracción de elementos del modelo, o sea el objetivo, las variables y los recursos, es realizada por el ser humano teniendo en cuenta cada palabra del texto con su respectivo contexto.
- Los seres humanos utilizan las siguientes reglas para armar las estructuras del modelo en su mente:

Para la Función Objetivo:

- Si en un subpárrafo clasificado como de Función Objetivo, se encuentra una variable y un número, entonces ese número corresponderá al aporte que cada unidad de la variable le hace a la función objetivo.

Para las Restricciones de Recursos:

- Si en un subpárrafo clasificado como Restricción, se encuentra un recurso, un número y una expresión que indique que el número limita al recurso, entonces ese número corresponderá al límite de dicho recurso y la expresión determinara el signo de la ecuación o inecuación según corresponda.
- Si en un subpárrafo clasificado como Restricción, se encuentra un recurso, un número, una variable y una expresión que indique que la variable consume al recurso, entonces ese número es lo que cada unidad de la variable consume del recurso.

Para las Restricciones de Variables:

- Si en un subpárrafo clasificado como Restricción, se encuentra un número, un tipo de variable y una expresión que indique limite, entonces este número restringe a la sumatoria de todas las variables y la expresión indica el signo de la ecuación o inecuación según corresponda.
- Si en un subpárrafo clasificado como Restricción, se encuentran una variable, un número y una expresión que indique limite, entonces este número limita la variable y la expresión indica el signo de la ecuación o inecuación según corresponda.
- Si en un subpárrafo clasificado como Restricción, se encuentra una o más variables acompañadas de números y expresiones que indique las relaciones entre las variables, entonces la expresión determina la relación matemática entre las variables del texto y el signo de la ecuación o inecuación.

Finalmente después de aplicar estas reglas el ser humano obtiene el modelo deseado.

La intención de este proyecto es traducir el proceso descrito anteriormente a términos computacionales con los cuales Dionisio pueda construir sus modelos, a continuación se presentan las conclusiones que obtenidas:

- A diferencia de los seres humanos, antes de dividir y clasificar, Dionisio determina el objetivo del problema como primer paso para afrontar la ambigüedad del lenguaje.
- Actualmente las computadoras no tienen la tecnología necesaria para entender el lenguaje a nivel humano por ello se hizo necesario utilizar conceptos de Procesamiento de Lenguaje Natural para llevar a cabo la división en “unidades” de texto que contuvieran como máximo un dato del modelo matemático, estas “unidades” se llaman subpárrafos.
- Para simular la forma de clasificar de un humano se crearon patrones en base al léxico y a expresiones utilizadas frecuentemente en los problemas de programación lineal, de acuerdo a estos y dependiendo del objetivo cada subpárrafo se clasifica utilizando conceptos de Lógica Difusa.
- Dionisio busca los elementos del problema a través de patrones que han sido contruidos de acuerdo al análisis pragmático del texto es decir teniendo en cuenta el contexto, tal como lo hace un ser humano.
- A partir de los elementos identificados el ser humano estructura el modelo en su cerebro agrupando los elementos según corresponda, dependiendo de la clasificación del texto y de las relaciones que en el encuentre.
- Para imitar la construcción del modelo del ser humano, Dionisio utiliza estructuras de información para cada elemento del modelo matemático, estas estructuras se cargan con base a los datos extraídos del texto aplicando reglas similares a las que aplica un ser humano.

En el campo académico hasta ahora la construcción de modelos matemáticos ha sido enseñada a través de imitación, ahora gracias a Dionisio se puede proponer una metodología de enseñanza de modelado, básicamente en esta metodología los modeladores principiantes deben llevar a cabo los siguientes pasos:

1. Leer el texto y dividirlo en párrafos.

2. Leer cada párrafo y dividirlo en subpárrafos.
3. Clasificar el subpárrafo según su contenido y su contexto de acuerdo a las partes de un modelo matemático de programación lineal.
4. Leer cada subpárrafo buscando los elementos del modelo según sea su clasificación.
5. Armar las estructuras del modelo matemático a partir de los elementos obtenidos y de la clasificación de los subpárrafos.
6. Construir el modelo matemático a partir de las estructuras.

Como Dionisio lleva a cabo un proceso similar, el usuario puede comparar su creación con la creación de Dionisio a través de las diferentes etapas del proceso.

8. RECOMENDACIONES

A partir de la construcción de Dionisio, han surgido las siguientes recomendaciones, para tener en cuenta a la hora de desarrollar un proyecto similar:

- Observar en detalle el modelo que se desea construir, con el fin de determinar, los elementos que la herramienta debe identificar y armar.
- Identificar como el ser humano realiza el proceso de modelado, como se mencionó anteriormente, esto puede llevarse a cabo a través de las técnicas utilizadas en la construcción de un Sistema Experto. En este aspecto, es importante detallar cada paso del proceso humano e intentar traducirlo en reglas de fácil implementación a nivel computacional.
- Crear de estructuras de información, que permitan manipular la información que proveen los elementos identificados en el texto, cabe mencionar, que en la mayoría de los casos, estos elementos no tienen un significado relevante, si no se tiene en cuenta su contexto.
- Estructurar el proceso de modelado de forma óptima en términos computacionales, aunque los pasos que lleva a cabo el ser humano para modelar, son importantes, no es necesario seguirlos a cabalidad, a nivel de máquina, el orden y complejidad de estos pueden variar.
- Crear una interfaz amigable al usuario, que permita fácilmente cargar información, y lo más importante, que la explicación del modelo o resultado por parte de la herramienta se presente en una forma fácil de asimilar para el usuario.

9. BIBLIOGRAFIA

BAUER KLAUS, BUTTEL IRMGARD, EBERHARD LUTZ, “Sistemas Expertos: Introducción a la Técnica y Aplicación”, 1ª Ed., Marcombo BOIXREU EDITORES, Barcelona – España, 1988.

DEL BRIO BONIFACIO MARTIN, SANZ MOLINA ALFREDO, “Redes Neuronales y Sistemas Difusos”, 2ª Ed., ALFAOMEGA GRUPO EDITOR– RA - MA, Madrid – España, 2002

DAVID W. ROLSTON, “Principios de Inteligencia Artificial y Sistemas Expertos”, 1ª Ed., Mc GRAW-HILL, México D.F., 1992.

FREDERICK S. HILLIER, GERALD J. LIEBERMAN, “Introducción a la Investigación de Operaciones”, 8ª Ed., Mc GRAW-HILL, México D.F., 2006.

MATHUR KAMLESH, SOLOW DANIEL, “Investigación de Operaciones: El Arte de la Toma de Decisiones”, 1ª Ed., PRENTICE - HALL, México D.F., 1996.

MOCHÓN, S. Modelos Matemáticos para todos los niveles. Cuadernos Didácticos. Vol. 9. Grupo Editorial Iberoamérica. México, D. F., 2000.

MOSKOWITZ HERBERT, WRIGHT GORDON, “Investigación de Operaciones”, 1ª Ed., PRENTICE - HALL, México D.F., 1982.

NILS J. NILSSON, “Inteligencia Artificial: Una Nueva Síntesis”, 1ª Ed., McGRAW – HILL, Madrid – España, 2001.

OVALLE DEMETRIO ARTURO, CAÑÓN RODRIGUEZ JAIRO, “Sistemas Inteligentes Artificiales y su Aplicación en Ingeniería”, 1ª Ed., Universidad Nacional de Colombia, Medellín – Colombia, 1997.

PAJARES MARTINSANZ GONZALO, SANTOS PEÑAS MATILDE, “Inteligencia Artificial e Ingeniería del Conocimiento”, 1ª Ed., ALFAOMEGA GRUPO EDITOR, México, 2006[1][1].

PRESSMAN ROGER, "Ingeniería del Software un Enfoque Práctico.", 2ª Ed., MC GRAW HILL, Madrid – España 2002

STUART RUSSELL, PETER NORVIG, "Inteligencia Artificial un Enfoque Moderno", 1ª Ed, PRENTICE HALL, México D.F., 1996

TAHA HAMDY A., "Investigación de Operaciones", 7ª Ed., PEARSON EDUCACION, México D.F., 2004.

10. CIBERGRAFIA

COLECCIÓN DE TESIS DIGITALES UDLAP PUEBLA – MEXICO. Capítulo 3. Lógica Difusa (En línea).

En http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/ramirez_r_o/capitulo3.pdf

DEPARTAMENTO DE COMPUTACION CINVESTAV MEXICO DF - MEXICO. Introducción a la Lógica Difusa (En línea). En: <http://delta.cs.cinvestav.mx/~gmorales/ldifll/ldifll.pdf>

EL PROFESIONAL DE LA INFORMACION – Revista Internacional Científica y Profesional – ESPAÑA. SOSA EDUARDO - Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones (En línea) http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actual_bases_tericas_y_aplicaciones_parte_i.html

TESIS DOCTORALES EN RED CATALUNYA - ESPAÑA. Capítulo 2. Conceptos Fundamentales de Lógica Difusa (En línea). En http://www.tesisenxarxa.net/TESIS_UPC/AVAILABLE/TDX-0207105-105056//04Rpp04de11.pdf

ANEXOS

INSTALACIÓN

La herramienta cuenta con un instalador que guíara paso a paso al usuario en el proceso de instalación. Además de presentarle los requerimientos de Dionisio en cuanto a software y hardware.

INSTRUCCIONES

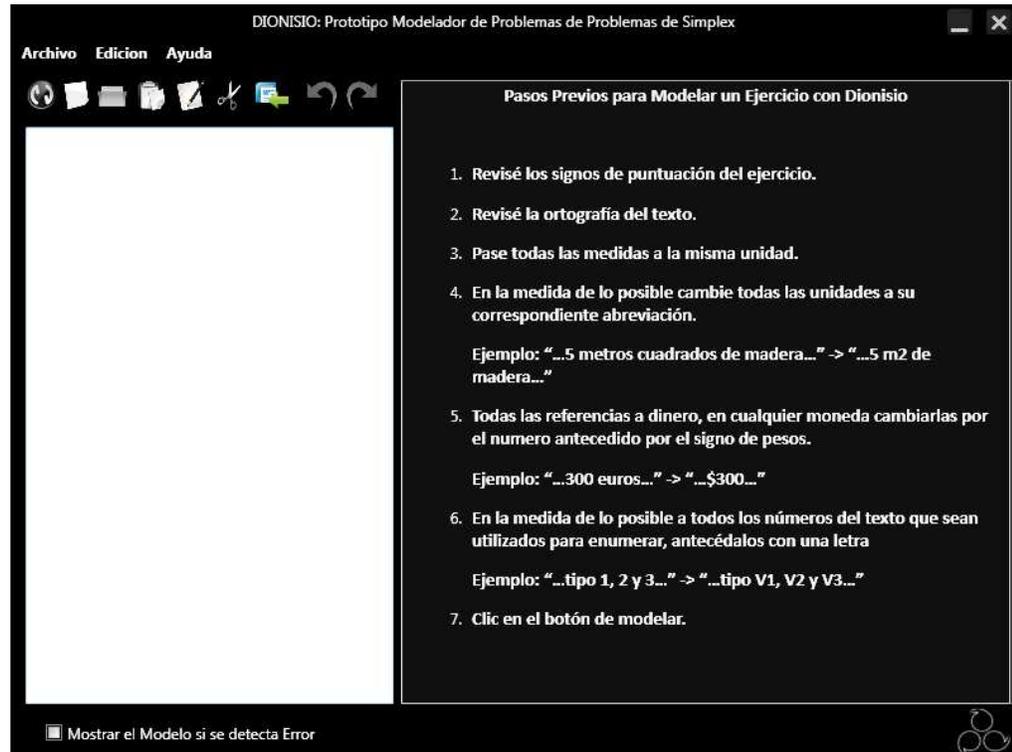
Las instrucciones de cómo utilizar a Dionisio son las siguientes:

- Digite el texto del problema.
- Verifique que el texto cumpla con las reglas establecidas en Pasos Previos para Modelar un Ejercicio con Dionisio.
- Establezca si desea que Dionisio le muestre el modelo aunque existan errores en el proceso chequeando la opción.
- Haga clic en modelar.
- Despliegue el tree –view para ver el modelo creado por Dionisio.

INTERFACES:

Como se puede ver en la siguiente figura (Figura 63), la primera interfaz, que se presentara al usuario es una ventana donde se pueden identificar cuatro partes:

Figura 63 - Interfaz Principal



- Barra de Herramientas: Ubicada en la parte superior en ella se pueden encontrar la siguientes opciones:

Archivo: En esta opción se presentan acciones para manejo de archivos y ejercicios entre estas acciones están:

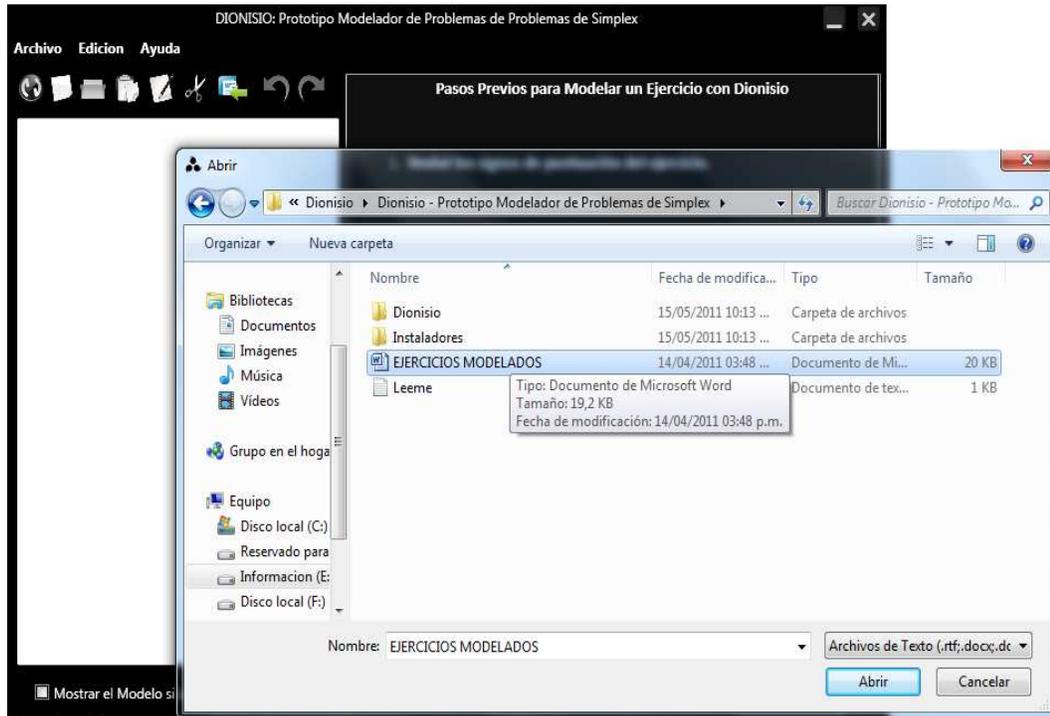
Nuevo: Esta opción permite limpiar el espacio de texto y el panel derecho donde se muestra el resultado con el fin de ingresar el texto de un ejercicio totalmente nuevo.

Modelar: Esta opción permite disparar el proceso de modelado del texto se puede acceder a ella desplegando este menú o a través de su acceso rápido.

Importar Texto: Esta opción permite cargar el texto que desea modelar a partir de archivos existentes en su computador, los archivos desde los cuales se puede cargar el texto son los de extensión doc, docx y txt.

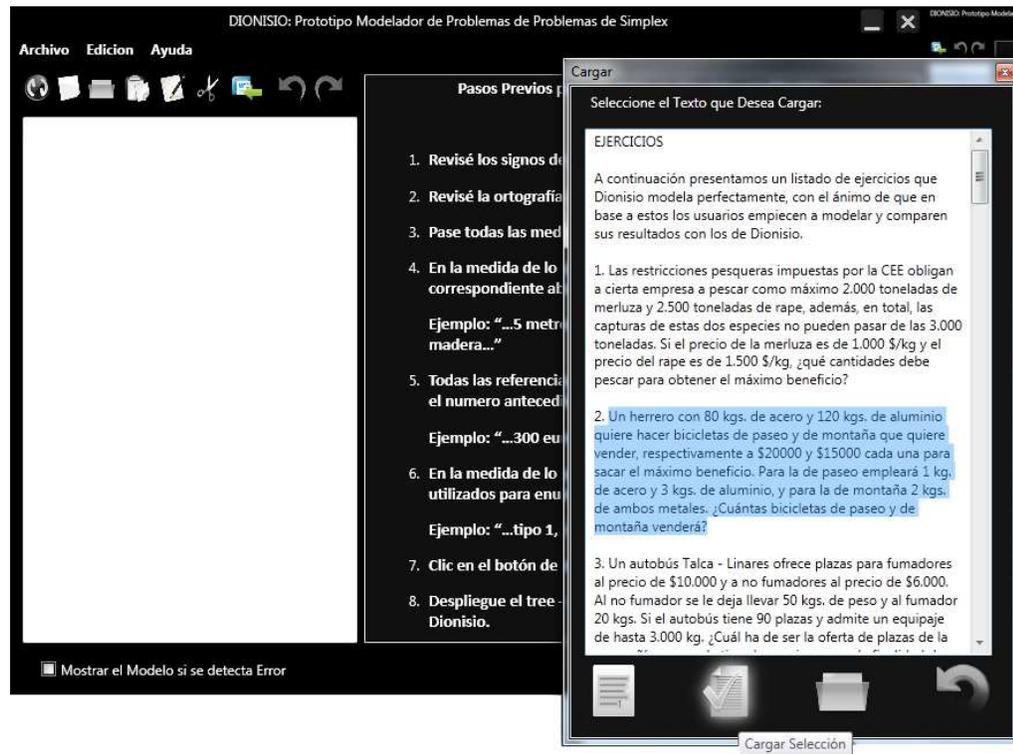
Lo primero que deberá hacer el usuario, como se muestra en la siguiente figura (Figura 64), es escoger el archivo que desea cargar.

Figura 64 - Interfaz Importar Texto



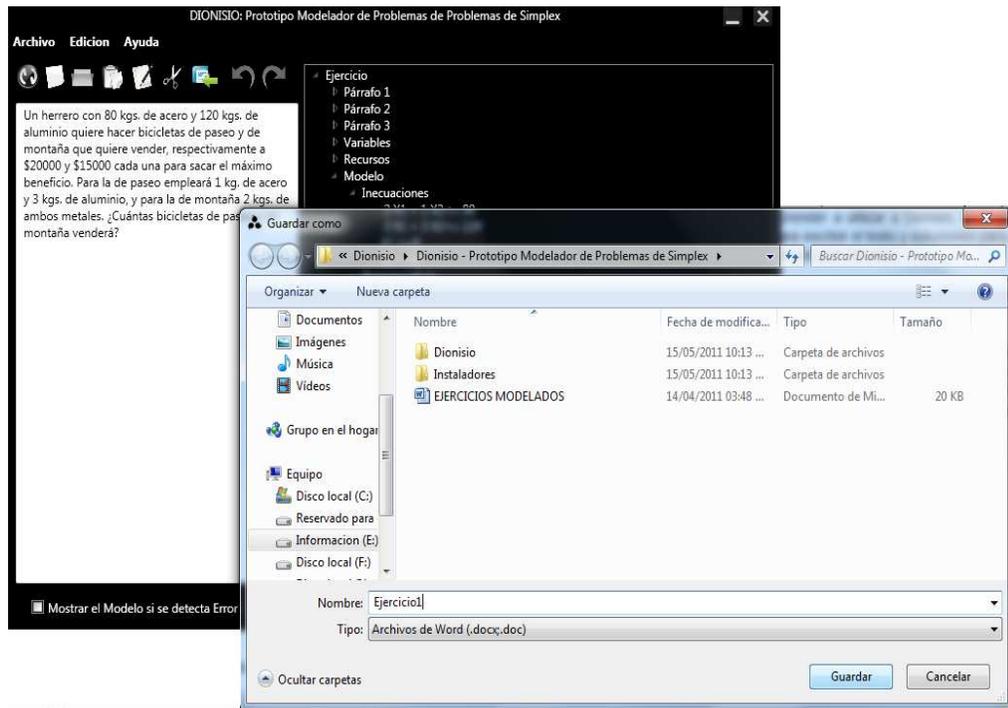
A continuación, el texto de este archivo se cargara en una ventana de Dionisio (Figura 65), donde el usuario podrá seleccionar el texto o el problema que desea cargar y modelar.

Figura 65 - Interfaz Cargar Selección



Exportar a Word: Esta opción permite crear un archivo de Word con el texto del problema y su respectivo modelo, si el ejercicio no está modelado, esta opción dispara el proceso de Dionisio y al final solicita al usuario una ruta donde guardar el archivo de Word (Figura 66).

Figura 66 - Interfaz Exportar Archivo



Salir: Esta opción permite salir de la aplicación

Edición: En esta opción se presentan acciones para edición del texto que se desea modelar entre estas se encuentran:

Deshacer: Esta opción permite deshacer las acciones que se realicen sobre el texto que desea modelar.

Rehacer Esta opción permite rehacer las acciones que se realicen sobre el texto que desea modelar.

Copiar: Esta opción permite copiar el texto que desea modelar.

Cortar: Esta opción permite cortar el texto que desea modelar.

Pegar: Esta opción permite pegar texto que haya sido copiado de otras partes del computador, como archivos de texto, exploradores de internet, o cualquier aplicación donde se maneje texto que permita ser copiado.

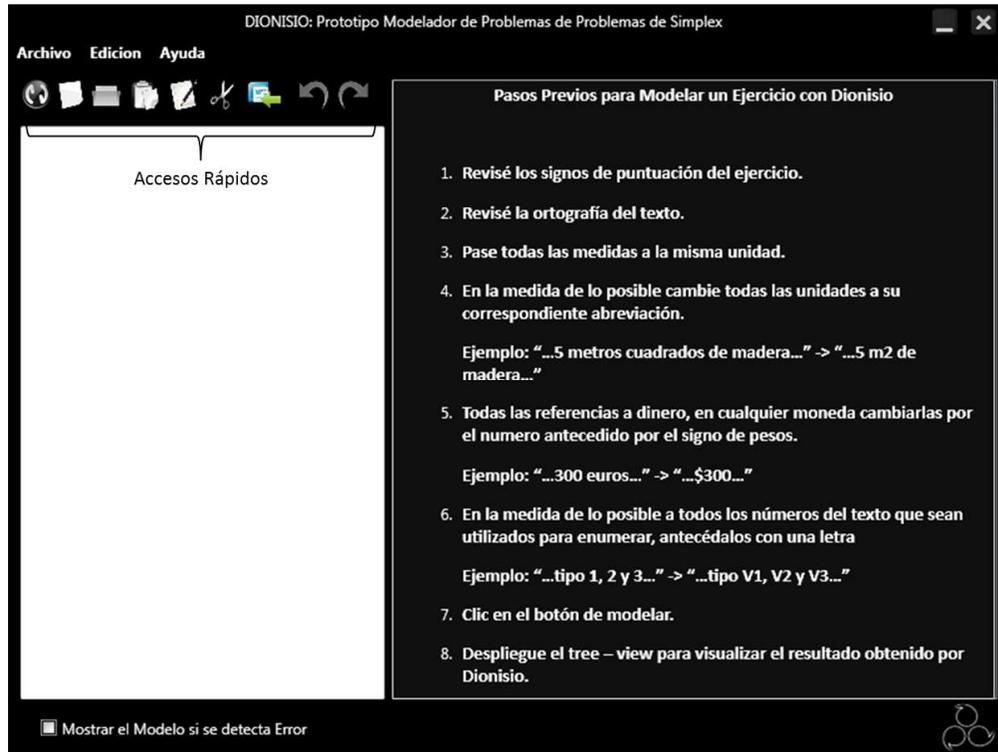
Ayuda: En esta opción se presenta información acerca de Dionisio, agrupadas en dos opciones:

Ver Ayuda: En esta opción se encontrara la ayuda que le guiara en el proceso de aprender a utilizar a Dionisio, descripción de interfaz, sugerencias para escribir el texto y soluciones para posibles errores en el modelo creado por Dionisio.

Acerca de: En esta opción se encontrara información acerca de los autores de Dionisio.

- Accesos Rápidos: En esta parte de la interfaz se presentan opciones de acceso rápido que permitirán al usuario utilizar las funcionalidades de Dionisio de una manera eficiente, estas funcionalidades, son las listadas en la barra de herramientas en los menús de Archivo y Edición (Figura 67).

Figura 67 - Interfaz Accesos Rápidos



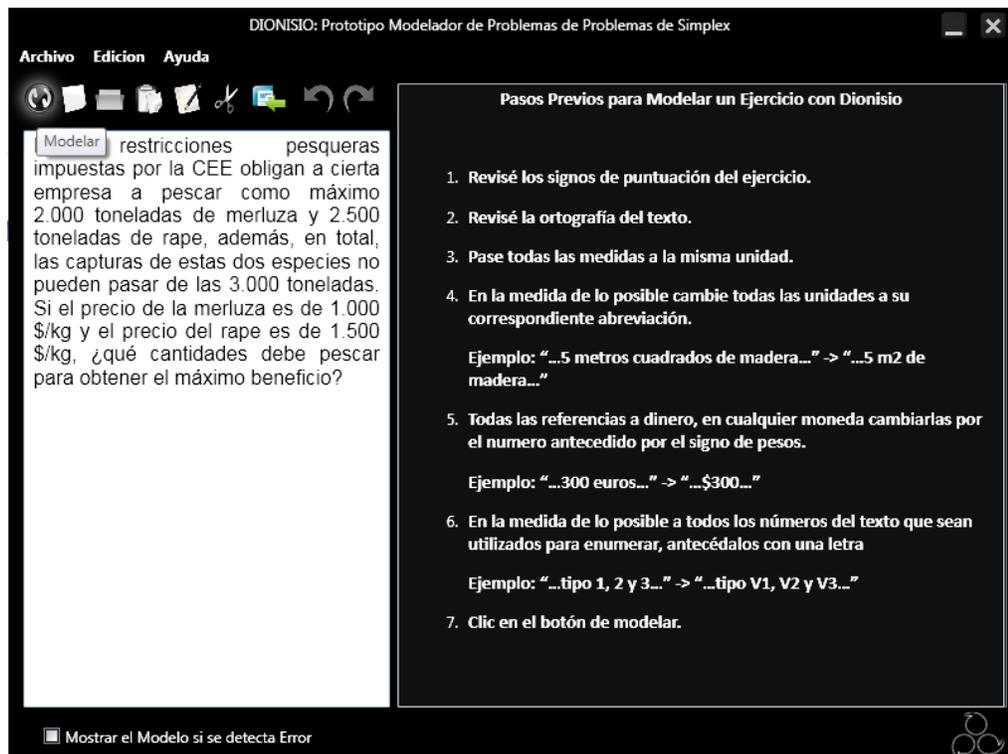
Entre estos se encuentran:

- Modelar.
 - Nuevo.
 - Importar Texto.
 - Pegar.
 - Copiar.
 - Cortar.
 - Exportar a Word.
 - Deshacer.
 - Rehacer.
- Espacio de Texto: Ubicada en la parte izquierda de la pantalla, aquí el usuario puede digitar, cargar, copiar o pegar el texto del problema que desea modelar.
 - Panel Derecho: Aquí se listara diferente información dependiendo de la opción escogida, inicialmente se muestra información del pre procesamiento que usted debe hacer del texto antes de dar clic en modelar,

posteriormente cuando se dispare el proceso de modelado y Dionisio termine, en este lugar se presenta el resultado del proceso de modelado de Dionisio en un tree – view que deberá ser desplegado para visualizar el resultado del proceso.

Después de pegar o digitar el texto en el espació correspondiente el usuario debe disparar el proceso de modelado a través de la opción modelar que se despliega en archivo o a través del acceso rápido modelar, como se muestra a continuación (Figura 68).

Figura 68 - Interfaz Modelar

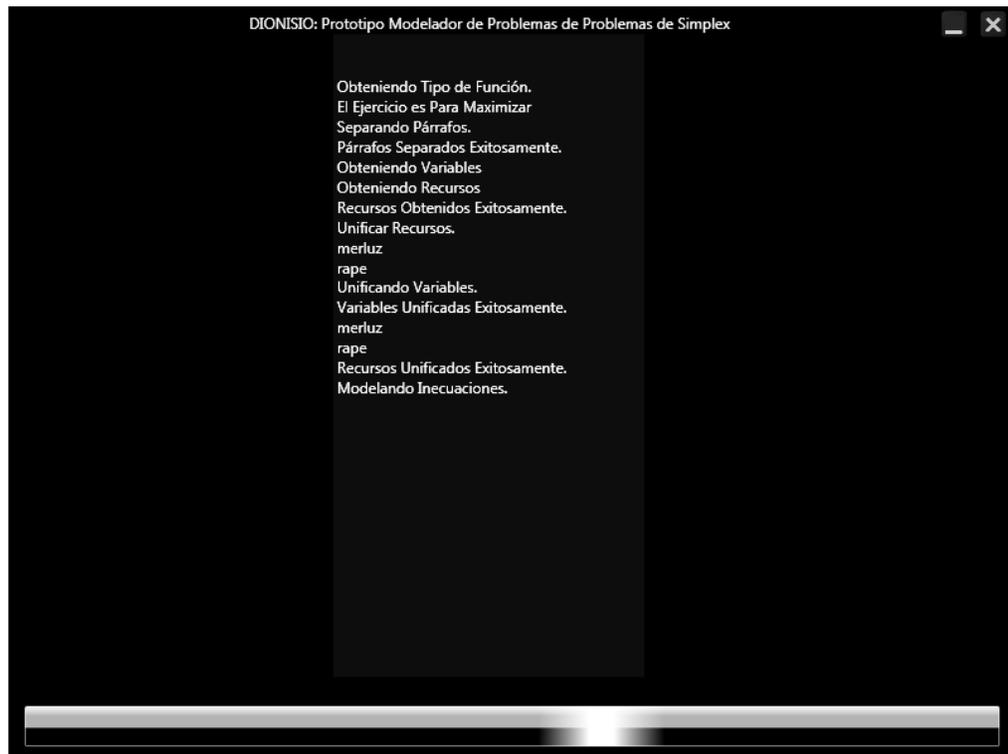


Además el usuario puede seleccionar si desea ver el modelo aunque haya errores en el proceso de modelado a través de una opción que debe chequear.

Después de que el usuario haya dado clic en modelar la siguiente interfaz (Figura 69), que a continuación se presenta, muestra en tiempo real como Dionisio va realizando el proceso de modelado, detallando los resultados de cada etapa desde

la obtención del objetivo hasta la construcción de las estructuras pasando por la división y clasificación de texto, además de la obtención de elementos del modelo.

Figura 69 - Interfaz Modelando Texto



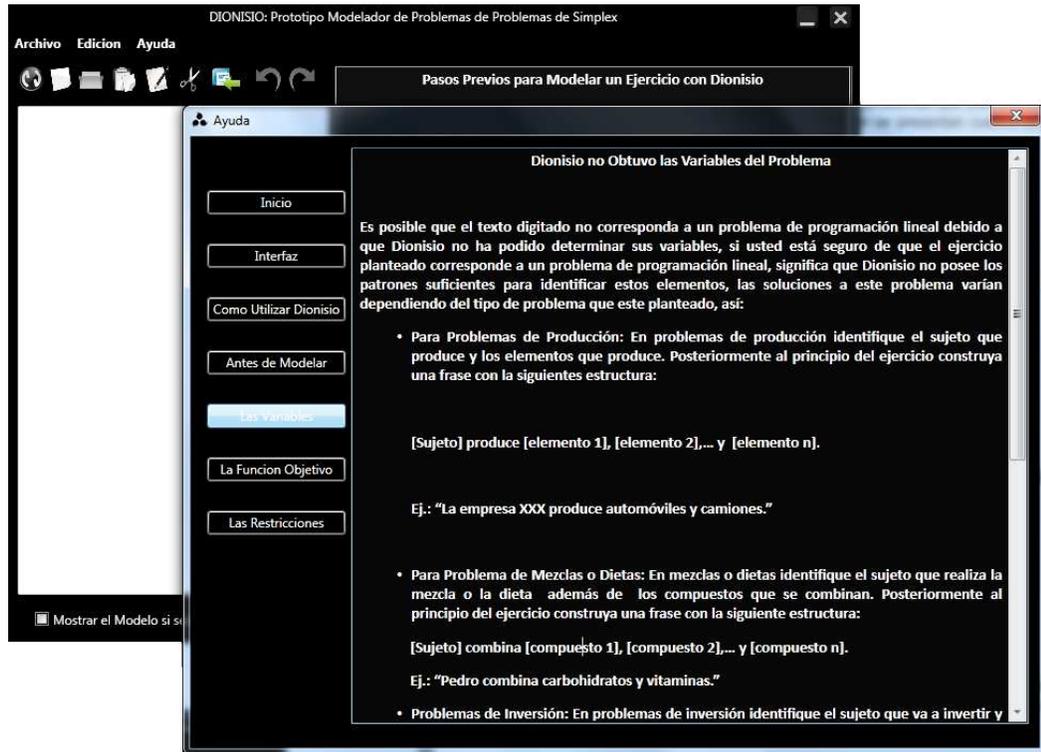
Finalmente, como se presenta en la siguiente imagen (Figura 70), el resultado se presenta en una interfaz similar a la inicial, la única diferencia es que en el lugar donde se listaban los pasos previos ahora se muestra una lista de párrafos y subpárrafos con su respectiva clasificación y sus respectivos elementos, las variables y los recursos encontrados con su respectiva nomenclatura en caso de las variables y las estructuras correspondientes al modelo final para que el usuario pueda comparar su proceso de modelado con el de Dionisio.

Figura 70 - Interfaz Presentación Modelo



Además, como se muestra a continuación (Figura 71), Dionisio presenta al usuario una ventana de ayuda donde están las principales sugerencias para utilizar Dionisio además de las posibles soluciones ante los problemas que pueden presentarse al ejecutar Dionisio, estas soluciones también se presentan cuando el proceso de Dionisio falla.

Figura 71 - Interfaz de Ayuda



EJERCICIOS

A continuación se presenta un listado de ejercicios que Dionisio modela perfectamente, con el ánimo de que en base a estos los usuarios empiecen a modelar y comparen sus resultados con los de Dionisio.

1. Las restricciones pesqueras impuestas por la CEE obligan a cierta empresa a pescar como máximo 2.000 toneladas de merluza y 2.500 toneladas de rape, además, en total, las capturas de estas dos especies no pueden pasar de las 3.000 toneladas. Si el precio de la merluza es de 1.000 \$/kg y el precio del rape es de 1.500 \$/kg, ¿qué cantidades debe pescar para obtener el máximo beneficio?
2. Un herrero con 80 kgs. de acero y 120 kgs. de aluminio quiere hacer bicicletas de paseo y de montaña que quiere vender, respectivamente a \$20000 y \$15000 cada una para sacar el máximo beneficio. Para la de paseo empleará 1 kg. de acero y 3 kgs. de aluminio, y para la de montaña 2 kgs. de ambos metales. ¿Cuántas bicicletas de paseo y de montaña venderá?
3. Un autobús Talca - Linares ofrece plazas para fumadores al precio de \$10.000 y a no fumadores al precio de \$6.000. Al no fumador se le deja llevar 50 kgs. de peso y al fumador 20 kgs. Si el autobús tiene 90 plazas y admite un equipaje de hasta 3.000 kg. ¿Cuál ha de ser la oferta de plazas de la compañía para cada tipo de pasajeros, con la finalidad de optimizar el beneficio?
4. Un comerciante acude al mercado popular a comprar naranjas con \$50.000. Le ofrecen dos tipos de naranjas: las de tipo A a \$50 el kg. y las de tipo B a \$80 el kg. Sabiendo que sólo dispone de su camioneta con espacio para transportar 700 kg. de naranjas como máximo y que piensa vender el kg. de naranjas tipo A a \$58 y el kg. de tipo B a \$90: ¿Cuántos kg. de naranjas de cada tipo deberá comprar para obtener máximo beneficio? ¿Cuál será ese beneficio máximo?
5. Un sastre tiene 80 m² de tela de algodón y 120 m² de tela de lana. Un traje requiere 1 m² de algodón y 3 m² de lana, y un vestido de mujer requiere 2 m² de cada una de las dos telas. Calcular el número de trajes y vestidos que debe confeccionar el sastre para maximizar los beneficios si un traje y un vestido se venden al mismo precio.

6. La fábrica LA MUNDIAL S.A., construye mesas y sillas de madera. El precio de venta al público de una mesa es de 2.700 y el de una silla 2.100. LA MUNDIAL S.A. estima que fabricar una mesa supone un gasto de 1.000 de materias primas y de 1.400 de costos laborales. Fabricar una silla exige 900 de materias primas y 1.000 de costos laborales. La construcción de ambos tipos de muebles requiere un trabajo previo de carpintería y un proceso final de acabado (pintura, revisión de las piezas fabricadas, empaquetado, etc.). Para fabricar una mesa se necesita 1 hora de carpintería y 2 horas de proceso final de acabado. Una silla necesita 1 hora de carpintería y 1 hora para el proceso de acabado. LA MUNDIAL S.A. no tiene problemas de abastecimiento de materias primas, pero sólo puede contar semanalmente con un máximo de 80 horas de carpintería y un máximo de 100 horas para los trabajos de acabado. Por exigencias del mercado, LA MUNDIAL S.A. fabrica, como máximo, 40 mesas a la semana. No ocurre así con las sillas, para los que no hay ningún tipo de restricción en cuanto al número de unidades fabricadas. Determinar el número de mesas y de sillas que semanalmente deberá fabricar la empresa para maximizar sus beneficios.

7. Un pastelero fabrica dos tipos de tartas T1 y T2, para lo que usa tres ingredientes A, B y C. Dispone de 150 kgs. de A, 90 kgs. de B y 150 kgs. de C. Para fabricar una tarta T1 debe mezclar 1 kgs. de A, 1 kgs. de B y 2 kgs. de C, mientras que para hacer una tarta T2 se necesitan 5 kgs. de A, 2 kgs. de B y 1 kgs. de C. a. Si se venden las tartas T1 a 1.000 la unidad y las T2 a 2.300. ¿Qué cantidad debe fabricar de cada clase para maximizar sus ingresos?

8. Una fábrica produce chaquetas y pantalones. Tres máquinas (de cortar, coser y teñir) se emplean en la producción. Fabricar una chaqueta representa emplear la máquina de cortar 1 hora, la de coser 3 horas y la de teñir 1 hora; fabricar unos pantalones representa usar la máquina de cortar 1 hora, la de coser 1 hora y la de teñir 0. La máquina de teñir se puede usar durante 3 horas, la de coser 12 y la de cortar 7. Todo lo que se fabrica es vendido y se obtiene un beneficio de \$8 por cada chaqueta y de \$5 por cada pantalón. ¿Cómo emplearíamos las máquinas para conseguir el beneficio máximo?

9. En una explotación agrícola con 25 Ha., pueden establecerse dos cultivos A y B. El beneficio de una Ha. de A es de 20000 ptas. y el de una Ha. de B de 30000 ptas. Las disponibilidades de trabajo de explotación son de 80 jornadas, una Ha. de A precisa 4 jornadas, mientras que una de B precisa sólo 2 jornadas. La subvención de la Unión Europea es de 5 euros por Ha. de A y de 10 euros por Ha.

de B, siendo la subvención máxima por explotación agrícola de 200 euros. Calcular el beneficio máximo.

10. SONY fabrica dos productos: el Walkman un radiocasete portátil y el Shader TV, un televisor en blanco y negro del tamaño de un reloj de pulsera. El proceso de producción de ambos productos se asemeja en que los dos necesitan un número de horas de trabajo en el departamento de electrónica, y un cierto número de horas de mano de obra en el departamento de montaje. Cada Walkman necesita 4 horas de trabajo de electrónica y 2 en el taller de montaje. Cada televisor necesita 3 horas de electrónica y 1 en montaje. Durante el actual período de producción se dispone de 240 horas en el departamento de electrónica y de 100 horas en el de montaje. Cada Walkman vendido supone un beneficio de \$7, mientras que para un televisor el beneficio unitario es de \$5. El problema de SONY es determinar la mejor combinación posible de Walkman y televisores que debe producir para alcanzar el máximo beneficio.

11. Un lago de montaña en un parque nacional tienen en la primavera de cada año dos especies de peces, S1 y S2. El peso promedio de cada pez en el lago es de 4 libras para S1 y de 2 libras para S2. Se dispone de dos alimentos: F1 y F2. Las necesidades promedio de un pez de especie S1 son de 1 unidad de F1 y 3 unidades de F2 diariamente. Las necesidades correspondientes de S2 son de 2 unidades de F1 y 1 unidad de F2. Si se cuenta con 500 unidades de F1 y 900 unidades de F2 por día, ¿cómo debe ser la cantidad de peces de cada clase para maximizar el peso de pescado que se pueda producir?

12. El administrador del sistema de suministro de agua de cierta ciudad debe hallar la manera de proporcionar por lo menos 10 millones de galones de agua potable (mgd.) por día. El agua se debe tomar de los depósitos locales o de una tubería que pasa hacia una población vecina. Los depósitos locales pueden suministrar 5 mgd., cantidad que no puede ser excedida. Debido a su tamaño, la tubería puede suministrar un máximo de 10 mgd. Además, debido a una cláusula contractual, de la tubería se debe bombear por lo menos 6 mgd. Por último, el costo del agua del depósito es de 300 dólares por cada mg y el costo del agua de la tubería es de 500 dólares por cada mg. ¿En qué forma puede el administrador minimizar el costo diario del agua?

13. Una campaña para promocionar una marca de productos lácteos se basa en el reparto gratuito de yogurt con sabor a limón o a fresa. Se decide repartir al menos 30.000 yogurt. Cada yogurt de limón necesita para su elaboración 0,5 gr. de un

producto de fermentación y cada yogurt de fresa necesita 0,2 gr. de ese mismo producto. Se dispone de 9 kgs. de ese producto para fermentación. El costo de producción de un yogurt de fresa es el doble que el de un yogurt de limón. ¿Cuántos yogurt de cada tipo se deben producir para que el costo de la campaña sea mínimo?

14. La empresa FORD lanza una oferta especial en dos de sus modelos, ofreciendo el modelo A a un precio de 1,5 millones de y el modelo B en 2 millones. La oferta está limitada por las existencias, que son 20 autos del modelo A y 10 del B. Por otra parte, para cubrir gastos de esa campaña, los ingresos obtenidos por la venta de ambos modelos debe ser al menos de 6 millones ¿Cuántos automóviles de cada modelo deberá vender para maximizar sus ingresos?

15. La compañía ESPECIAS INDIAN S.A., tiene un stock limitado de dos hierbas que se utilizan en la producción de aderezos. INDIAN usa los dos ingredientes, HB1 y HB2, para producir curry y pimentón. El departamento de mercadotecnia informa que aunque la empresa puede vender todo el pimentón que pueda producir, sólo puede vender hasta un máximo de 1500 botellas de curry. Se determinó que una botella de curry necesita 5 onzas de HB1 y 3 de HB2, en tanto que la botella de pimentón requiere 2 onzas de HB1 y 3 de HB2. La compañía dispone de 10000 onzas del ingrediente HB1 y 8500 de HB2. Determine la producción de especias que maximice el ingreso de la Empresa si una botella de curry se vende a \$2750 y una y una botella de pimentón a \$1300.

16. La compañía de muebles Grant fabrica mesas y sillas. Por cada silla se necesitan 20 pies de madera y 4 horas de mano de obra. Por cada mesa se necesitan 50 pies de madera pero sólo 3 horas de mano de obra. El fabricante dispone de 3300 pies de madera y de una planta laboral que le puede proporcionar 380 horas de mano de obra. Por último, el fabricante ha determinado que se obtiene una ganancia neta de 3 dólares por cada silla vendida y una ganancia de 6 dólares por cada mesa vendida. Por comodidad, se supondrá que los materiales adicionales requeridos (clavos, barniz, etc.) se tienen en cantidades suficientes y no se toman en cuenta para costos finales. En estas condiciones, ¿cuántas mesas y cuántas sillas deberá fabricar la compañía con el fin de maximizar su ganancia, suponiendo que se venden todos los artículos producidos?

17. La Wyndor Glass Co., produce puertas y ventanas en tres plantas A, B y C. La producción de una puerta requiere de 1 hora en la planta A y de 3 horas en la

planta C. La producción de una ventana requiere de 2 horas en la planta B y de 2 horas en la planta C. La disponibilidad máxima de horas para las plantas A, B y C, es respectivamente de 4, 12 y 18 horas. Una puerta produce una ganancia neta de \$3 y una ventana produce una ganancia de \$5. Determine cuántas puertas y cuántas ventanas se deben fabricar semanalmente con el fin de maximizar las ganancias.

18. Una fábrica ofrece al mercado cuatro productos p1, p2, p3 y p4, Que requieren para su elaboración de una materia prima de la cual hay una disponibilidad diaria de 180 Kg, un espacio total disponible para almacenamiento de 230 m³, y para la producción se dispone de 5 horas de trabajo. Se sabe que p1 deja una ganancia de \$5 y requiere 2 Kg. de materia prima, 2 m³ de espacio y 0.06 horas de trabajo, p2 deja una ganancia de \$6,5 y requiere 2 Kg. de materia prima, 2,5 m³ de espacio y 0.03 horas de trabajo, p3 deja una ganancia de \$5 y requiere 1,5 Kg. de materia prima, 2 m³ de espacio y 0.1 horas de trabajo y p4 deja una ganancia de \$5,5 y requiere 4 Kg. de materia prima, 1,5 m³ de espacio y 0.06 horas de trabajo. ¿Cuántas unidades de cada producto son necesarias a fin de maximizar las utilidades?

19. Una compañía desea elaborar tres productos P1, P2 y P3 con \$1.5 millones. P1, P2 y P3, requieren respectivamente 30, 3 y 15 m³ de espacio por unidad. Y el espacio total disponible es de 300000 m³. Los productos P1, P2 y P3, cuestan respectivamente \$1200, \$4500 y \$15000. ¿Qué cantidad debe adquirirse de cada producto si el precio de venta de P1, P2 y P3, es respectivamente de \$1500, \$6000 y \$21000, para obtener el máximo beneficio?

20. La compañía Everfit produce prendas de hombre y de mujer, para lo que dispone de los siguientes materiales: 160 m² de algodón, 110 m² de seda y 150 m² de lana. Una prenda de mujer requiere 2 m² de algodón, 1 m² de seda y 1 m² de lana. Una prenda de hombre requiere 1 m² de algodón, 2 m² de seda y 3 m² de lana. Si una prenda de mujer deja una utilidad de \$9000 y una de hombre deja \$15000. ¿Cuántas prendas de hombre y cuántas de mujer debe confeccionar la empresa para obtener la máxima utilidad?