UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERIA EN CIENCIAS Y SISTEMAS

PROTOTIPO DE SISTEMA EXPERTO LEGISLATIVO: VERIFICACIÓN DE CONSTITUCIONALIDAD O INCONSTITUCIONALIDAD EN INICIATIVAS DE LEY, BASADO EN LA CONSTITUCIÓN DE LA REPÚBLICA DE GUATEMALA.

DEL ARTÍCULO 1 AL 10

MÓNICA JOSÉ BAUTISTA VÁSQUEZ

Asesorada por Inga. Marcela Elizabeth Velásquez Miranda

Guatemala, noviembre de 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PROTOTIPO DE SISTEMA EXPERTO LEGISLATIVO: VERIFICACIÓN DE CONSTITUCIONALIDAD O INCONSTITUCIONALIDAD EN INICIATIVAS DE LEY, BASADO EN LA CONSTITUCIÓN DE LA REPÚBLICA DE GUATEMALA. DEL ARTÍCULO 1 AL 10

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MÓNICA JOSÉ BAUTISTA VÁSQUEZ

ASESORADA POR INGA. MARCELA ELIZABETH VELÁSQUEZ MIRANDA
AL CONFERIRSE EL TITULO DE

INGENIERA EN CIENCIAS Y SISTEMAS

Guatemala, noviembre de 2005

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Inga. Ricardo Morales Prado
EXAMINADOR	Inga. Elizabeth Domínguez Alvarado
EXAMINADOR	Ing. Juan Alvaro Díaz Ardavin
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROTOTIPO DE SISTEMA EXPERTO LEGISLATIVO:
VERIFICACIÓN DE CONSTITUCIONALIDAD O INCONSTITUCIONALIDAD
EN INICIATIVAS DE LEY, BASADO EN LA CONSTITUCIÓN DE LA REPÚBLICA DE GUATEMALA.
DEL ARTÍCULO 1 AL 10

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 23 de febrero de 2004.

Mónica José Bautista Vásquez

AGRADECIMIENTOS

A mis amigos: Carlitos, Oneida, Ervin, Miltón Orozco, Mercedes, Ronald, Mario, Otto, Mónica, Julio, por compartir conmigo sus sueños, pasar conmigo desvelos, alegrías, tristezas y por brindarme una amistad sincera, al verlos a ustedes veo que aun existen hombres y mujeres que pueden brindar su corazón a manos llenas.

A los movimientos Halito y Koinonia por encontrar ahí amigos, valores y un verdadero amor por la causa de Dios.

A Carlitos porque ha sido más que un amigo, por saberme escuchar, comprender, darme su amor incondicional y darle el valor agregado a mi vida.

A la iglesia Jesucristo es el Señor Zona 21 e Iglesia de Dios Jesucristo es Señor San Pedro Sac., S. M. por ser parte del cuerpo de Cristo y por darme a conocer la verdad de mi Salvador, además de proporcionarme su amistad y amor.

A Inga. Marcela por asesorarme y permitir concluir este trabajo de graduación, así mismo, a todos los catedráticos de la Universidad de San Carlos de Guatemala, por contribuir en mi educación y formar en mí una profesional integra.

Al Ministerio de educación, especialmente al personal que trabaja en DIGEPA y la UDI, por creer en mi y apoyarme en cada paso que he emprendido.

A mi abuela Vita por ser de gran ayuda y en general a toda mi familia por amarme, comprenderme y aconsejarme.

A mis hermanos, Carlos por ser más que un padre en los años que más lo necesite y apoyarme durante la carrera, a Ronald por darme su amor y alegría, a Melany y su familia, porque se dan a manos llenas para que nunca falte nada, a Robertito por impartirme su conocimiento sin condición alguna y hacer posible esta tesis, además de ser un ejemplo vivo de la búsqueda incesante de Dios y por último a la más pequeña de mis hermanos pero que ocupa un gran lugar en mi corazón, Maria Alejandra por ser muchas veces madre y confidente; a todos mis hermanos los amo, los bendigo y les agradezco su apoyo.

A mi padre porque con su esfuerzo, dedicación y diligencia a provisto lo necesario para cumplir este sueño, siendo su vida entera una inspiración a seguir adelante cada día.

A mi madre por ser una gran mujer, que ha sembrado lo mejor en mi, que con su ejemplo me ha enseñado más que mil palabras y hoy la puedo llamar "mujer virtuosa".

Y mi vida entera doy en agradecimiento a mi padre celestial, por ser el divino maestro que en su plan perfecto ha guiado y ordenado todo mi caminar, si no fuese por su dulce y tierna voz guiando mi camino, mi vida no tendría sentido.

ÍNDICE

ÍNDICE DE ILUSTRACIONES

FIGURAS

GLOSARIO

Antinomia	Contradicción entre dos leyes.
Aprendizaje automático	Estudia como construir programas que mejoren automáticamente con la experiencia.
Aprendizaje automático por refuerzo	Define un objetivo para el algoritmo de aprendizaje y premia o castiga al sistema a partir de las acciones que realice.
Constitucional	Es cuando una ley, reglamento, iniciativa de ley, etc., esta apegada a los principios que la constitución enmarca.
EHSIS	Lenguaje basado en clips para generación de sistemas expertos.
Equiparación	Proceso emparejamiento, donde se considera a alguien o algo igual o equivalente a otra persona o cosa.
Inconstitucional	Opuesto a lo que dice la Constitución de la República.
Iniciativa de ley	Es cuando las personas que tienen el derecho de iniciativa de ley proponen una nueva ley o la reforma de una.
Inteligencia Artificial	Es la rama de la ciencia de la computación que centra sus esfuerzos en la consecución de sistemas inteligentes
Lógica de primer orden	Es uno de los formalismos más utilizados para representar conocimiento en Inteligencia Artificial. La lógica cuenta con un lenguaje formal mediante el cual

	es posible representar fórmulas llamadas axiomas que permiten describir fragmentos del conocimiento, y, además, consta de un conjunto de reglas de inferencia que aplicadas a los axiomas, permiten derivar nuevo conocimiento.
Marco	Es una colección de atributos o slots que poseen determinados valores y que describen una entidad del mundo.
Motor de inferencia	Es la aplicación informática que controla cuándo y cómo se debe aplicar la información contenida en la base de conocimientos para responder a los problemas planteados.
Paradigma	Patrones de pensamiento, universalmente, reconocidos por algún grupo de profesionales o expertos en alguna materia en cuestión.
Prototipo	Ejemplar original o primer molde en que se fabrica una figura, sistema u otra cosa.
Pseudocódigo	Se considera un primer borrador, dado que el pseudocódigo tiene que traducirse, posteriormente, a un lenguaje de programación.
Razonamiento basado en casos	Generación de soluciones a problemas actuales respecto de la base de planes pasados, los cuales se almacenan como casos en una memoria, el caso cuyas características se asemejen más a las situación actual, se le aplican estrategias que lo adapten a las necesidades actuales.
Sistema abierto	Interactúa, constantemente, con el ambiente en forma dual, o sea, lo influencia y es influenciado. El sistema abierto puede crecer, cambiar, adaptarse al ambiente

	y hasta reproducirse bajo ciertas condiciones ambientes.
Sistema experto	Sistema que procesa conocimientos e indica decisiones a tomar en la resolución de determinados problemas, razonando sus propios procesos con la explicación de cómo y por qué ha llegado a una conclusión.

RESUMEN

Cuando se presenta una iniciativa de ley al Congreso, una de las revisiones que se realizan a la misma es compararla con la Constitución de la República de Guatemala, si se encuentra que contradice a alguna de las premisas de la Constitución de la República de Guatemala, la iniciativa de ley es rechazada porque se le declara inconstitucional, de lo contrario, la iniciativa de ley es constitucional.

En este trabajo de graduación se da a conocer la construcción de un prototipo de sistema experto legislativo, el cual proporciona una solución a este problema, siendo su función, procesar conocimientos e indicar las decisiones que se deben tomar para la resolución del mismo, dando a conocer el proceso y el fundamento a través del cual se ha llegado a esta conclusión.

La representación del conocimiento por Marcos es la metodología utilizada en el Prototipo de Sistema Experto Legislativo descrito en este trabajo de graduación, algunas de sus ventajas son las siguientes: se maneja una estructura conceptual, lo cual es de suma importancia, dado que un concepto necesita que se incluya entre sus partes al concepto subordinado y que actúe de superordinado, esto permite armar una red de relaciones entre los conceptos; inferir en base a los atributos de aquellos conceptos que se encuentren en una jerarquía de orden superior, sin tener que estar especificando los atributos uno por uno, siendo definido como herencia de valores por defecto, lográndose así, ahorrar tiempo y trabajo.

En el capítulo 3, de los artículos 1 al 10 de la Constitución de la República de Guatemala se obtienen los conceptos y atributos de los mismos, luego se representan por medio de marcos y se muestra la relación que existe entre uno y otro concepto. Al estar claramente definidas las reglas y relacionados los conceptos se procede a la construcción del Prototipo de Sistema Experto Legislativo, mostrándose algunos ejemplos de cómo funciona el mismo, todo esto es explicado en el capítulo 4.

Habiéndose terminado de explicar el funcionamiento del Prototipo de Sistema Experto Legislativo, en el capitulo 5 se presentan los avances logrados y sugerencias para poder continuar con la construcción del mismo.

OBJETIVOS

General

Crear un Prototipo de Sistema Experto Legislativo que verifique si una propuesta de ley es constitucional o inconstitucional y dicte un juicio, basado en la Constitución Política de la República de Guatemala de 1985, restringido del artículo 1 al 10.

Específicos

- Analizar el problema para saber hacia dónde se desea llegar.
- Extraer el conocimiento de la Constitución de la República.
- Representar el conocimiento por medio de Marcos.
- Diseñar el Prototipo de Sistema Experto Legislativo.
- Construir el Prototipo de Sistema Experto Legislativo.
- Proporcionar guías y pautas para continuar con la construcción del Sistema Experto Legislativo.

INTRODUCCION

Los Sistemas Basados en Conocimiento permiten analizar, dar solución o ayudar a la toma de decisión de determinados problemas, aplicando, para esto, un razonamiento similar al que aplicaría un experto en esa materia, al resolver el mismo problema. Dado a esto, los Sistemas Basados en Conocimiento, han venido a ser de gran beneficio a los expertos, en el área en que estos se desempeñan.

La incorporación de estos Sistemas a diferentes áreas, se incrementa día a día, por ejemplo; en la medicina, en la música, en el negocio y como es el caso de este trabajo de graduación, en el derecho. Los sistemas expertos en el derecho se han especializado en diferentes ramas como lo son: laboral, civil, penal, tributario, etc., pero el tema del presente trabajo es enfocado a la rama del Derecho Legislativo, siendo, este sistema experto, una herramienta que ayude al legislador a la toma de decisiones.

El Legislador utiliza como base para la toma de decisión respecto a una Propuesta de Ley, La Constitución Política de La República de Guatemala de 1985, comparando los artículos de La Propuesta de Ley con respecto a aquellos establecidos en La Constitución de La República, aprobando, de esta manera, aquellos que no sean contradictorios con los artículos de La Constitución Política de La República de Guatemala.

El Prototipo de Sistema Experto Legislativo tiene en su base de conocimientos todas las premisas de los artículos 1 al 10, siendo estos unos pocos artículos de los que se compone la parte Dogmática de La Constitución

de La República de Guatemala, siendo esta parte dogmática aquella en donde se establecen los principios, creencias y, fundamentalmente, los derechos humanos, tanto individuales como sociales que se le otorgan al pueblo como sector gobernado frente al poder público.

1 MARCO TEÓRICO

1.1 Introducción

En la actualidad existe una gran variedad de aplicaciones orientadas a un campo específico de la inteligencia artificial, entre las que se cuentan los sistemas expertos. Existen antecedentes de aplicaciones de este tipo para el área del derecho, es el caso de este trabajo de graduación, en el cual se trata específicamente el área legislativa del derecho y su interacción con la inteligencia artificial. Con lo cual este proyecto de graduación se denomina Sistema Experto Legislativo.

Para poder entender la funcionalidad esperada del Sistema Experto Legislativo de este trabajo de graduación, es importante tener un conocimiento general que ayude a entender de mejor manera nuestro propósito, es decir, es necesario conocer acerca de: la Constitución de la República de Guatemala, su división y la razón por la que el legislador se basa en ella para tomar una decisión con respecto a una iniciativa de ley, la representación de marcos como la metodología a utilizar para construir el sistema experto, así como los antecedentes que dieron origen a este trabajo de graduación.

1.2 Constitución Política de la República de Guatemala

Cuando se presenta una iniciativa de ley al Congreso, una de las revisiones que se realizan a la misma, es compararla con la Constitución de la República de Guatemala, si se encuentra que contradice a alguna de las premisas de la Constitución de la República de Guatemala, la iniciativa de ley

es rechazada porque se le declara Inconstitucional, es decir, atenta contra los principios, normas e instituciones que la carta magna prescribe; de lo contrario, la propuesta de ley es Constitucional, es decir, esta apegada y acorde con los principios que la Constitución establece.

1.2.1 Concepto

Es la Suprema Ley alrededor de la cual giran todas la demás leyes de la República. Es la Ley fundamental que sirve para establecer los principios y los derechos de los guatemaltecos y para establecer la organización jurídica y política del Estado de Guatemala. Se dice que es la Ley suprema de Guatemala, porque todas las normas contenidas en la Constitución pueden ser desarrolladas por otras normas y otras leyes, pero nunca pueden ser contrariadas o tergiversadas, es decir, que sobre la Constitución no existe otra disposición o ley superior. (14 – 35,37)

1.2.2 División de la Constitución Política de la República de Guatemala

Para su mayor comprensión la Constitución se divide en 3 grandes partes:

1.2.2.1 La parte dogmática

Es aquella en donde se establecen los principios, creencias y fundamentalmente los derechos humanos, tanto individuales como sociales, que se le otorgan al pueblo como sector gobernado frente al poder público como sector gobernante, para que este último respete estos derechos. Esta

parte dogmática se encuentra contenida en el titulo I y II de la Constitución, desde el Preámbulo y del artículo 1 al 139.

1.2.2.2 La parte orgánica

Es la que establece cómo se organiza Guatemala, la forma de organización del poder, es decir las estructuras jurídico-políticas del Estado y las limitaciones del poder público frente a la persona, o sea a la población. Esta parte Orgánica se encuentra contenida en los títulos III, IV y V de la Constitución, del artículo 140 al 262.

1.2.2.3 La parte práctica

Es la que establece las garantías y los mecanismos para hacer valer los derechos establecidos en la Constitución y para defender el orden constitucional. Esta parte Práctica se encuentra contenida en el Titulo VI y VII de la Constitución, del artículo 263 al 281.

Debido a que cualquier ley o norma no puede contrariar o tergiversar a las normas de la constitución y porque no existe una ley superior a la Constitución es por ello que el legislador utiliza la constitución para detectar antinomias o inconstitucionales en el proyecto de ley.

1.3 Sistema experto

Es un sistema que procesa conocimientos e indica decisiones a tomar en la resolución de determinados problemas, razonando sus propios procesos con la explicación de cómo y por qué ha llegado a una conclusión. Es decir, se trataría de sistemas que, a partir de ciertas informaciones provistas por un

asesor, permiten resolver problemas en un dominio específico, mediante la simulación de los razonamientos que los expertos del sistema harían utilizando los conocimientos adquiridos.

1.3.1 Componentes de un sistema experto

Un Sistema Experto debe estar compuesto de una serie de elementos o unidades lógicas que puedan ser desarrolladas, creadas y modificadas de forma autónoma, estos son:

1.3.1.1 Base de conocimientos

Es el componente que define el Sistema Experto al ser el depósito estructurado de toda su información. Esta se desglosa en:

- Conocimiento descriptivo: información sobre hechos, situaciones, datos, normas, sentencias, doctrinas, etc.
- Conocimiento procedimental: integrado por reglas de producción heurística del tipo “si...entonces”.

1.3.1.2 Motor de inferencia

Es una aplicación Informática que se encarga de la correcta aplicación de las reglas de inferencia en las deducciones automáticas que realiza el Sistema Experto.

Controla cuándo y cómo se debe aplicar la información contenida en la base de conocimientos para responder a los problemas planteados.

1.3.1.3 La interfase

Permite al usuario comunicarse con el Sistema Experto y a su vez permite al ingeniero del conocimiento desarrollar el Sistema Experto actualizándolo, modificándolo o revisando la base de conocimientos.

1.4 Sistemas expertos y derecho

Entre los Sistemas Expertos más notorios de nuestros días se encuentran aquellos que se han especializado en áreas de estudio tales como: la medicina, música, genética, educación y Derecho.

Han proliferado en estos años una serie de proyectos y prototipos de Sistema Expertos jurídicos en materias como, liquidaciones tributarias, cálculo de indemnizaciones por accidentes laborales o de tráfico, predicción de las consecuencias jurídicas de impactos medioambientales, condiciones de adquisición de la nacionalidad y Derecho de familia, en concreto matrimonio y divorcio.

El enfoque de este proyecto va dirigido hacia la construcción de un Sistema Experto Legislativo; para lograr una comprensión adecuada de lo que se pretende con este sistema, es necesario dar a conocer una serie de sistemas similares cuya funcionalidad puede guiar hacia la definición del sistema esperado.

1.4.1 Antecedentes

1.4.1.1 SHYSTER

Creado por Popple J. en su doctorado en Inteligencia Artificial y Derecho. SHYSTER fue desarrollado en Estados Unidos, este consiste en un sistema de casos legalmente basados. Este sistema proporciona consejos a través de la relación de casos similares.

El sistema experto legal SHYSTER utiliza vectores de atributos-valores simples con valores (si, no, desconocido) que conforman las características del dominio a representar. Si bien el esquema de representación es sencillo, la importancia del mismo es que permite testear la metodología desarrollada por Popple en diferentes dominios. SHYSTER no usa ponderaciones para los atributos, sin embargo, Popple notó en experimentos que el uso de pesos introducía un comportamiento más inteligente.

Popple indicó que los factores que representan las características importantes del dominio no necesitan ser derivados de un análisis doctrinal. Por otra parte, aclaró que un sistema experto legal predictivo podría llegar a utilizar atributos subjetivos como parte del proceso de decisión judicial.

El sistema Shyster intenta utilizar mecanismo sencillo de razonamiento basado en casos, a través de esquemas de representación simples y procedimientos de inferencia estadísticos.

Shyster fue desarrollado siguiendo los pasos del proyecto “Solmon” que fue puesto en marcha por la facultad de derecho de la Universidad de Nueva

York en 1995. El proyecto en sí tenía el objetivo de desarrollar un software capaz de decidir casos sin la obligatoria referencia hacia el jurado.

1.4.1.2 FINDER

Desarrollado en Australia, FINDER un sistema experto capaz de dar solución a un conjunto limitado de problemas legales aportando un razonamiento en forma de opinión legal y analizando los casos mediante un sistema de descripción factual. “Finder”, básicamente, pregunta una serie de cuestiones a las que se responde con un “sí” o un “no”; tras este cuestionario entra en funcionamiento el mecanismo computacional del sistema que, a través de algoritmos, elige un resultado para el caso en cuestión tomando como referencia el caso más próximo entre los que se encuentran almacenados dentro de la base de conocimiento. Precisamente la referencia a este caso es la que confiere validez para el resultado anunciado.

1.4.1.3 HYPO

Desarrollado en Australia, HYPO emplea hechos para construir argumentos válidos a favor y en contra de una determinada posición. El sistema, al formular estos argumentos, identifica las debilidades de los mismos y las aprovecha para elaborar argumentos que puedan ser empleados en la oposición. Finalmente, tras este proceso, genera nuevos argumentos reforzados y modificados, para la posición original, a través del uso de hipótesis. Utilizando toda esta cadena argumental, el sistema puede decidir qué posición es la que obtiene la victoria.

Para el desarrollo del sistema se utiliza esquemas de representación basados en marcos.

El sistema experto legal HYPO utiliza un mecanismo de inferencia conocido como “3-ply-argument”. El mismo presenta el siguiente esquema:

1. Una recomendación para una de las partes citando casos.
2. Una respuesta para la otra parte con casos de contraejemplo.
3. Una refutación de la primera de las partes de los contraejemplos de la segunda.

En la teoría elaborada por HYPO, la determinación de las características salientes de un precedente y los precedentes que son significativos en un argumento dependen del contexto: los hechos del problema, los hechos y resultados de otros casos, el punto de vista del operador judicial, y el rol contextual del caso dentro del argumento.

1.4.1.4 LEAD

"Legal Advisor" (LEAD), cuyo objeto es proveer asesoramiento a los operadores judiciales en el proceso de individualización de la pena. El sistema tiene un comportamiento predictivo, con características adaptativas y explicativas.

A partir de las circunstancias objetivas y subjetivas consideradas relevantes para un determinado caso, LEAD es capaz de producir una recomendación de la pena. Asimismo, LEAD genera una justificación basada en jurisprudencia, que puede ser utilizada por los jueces como parte de la fundamentación de la sentencia.

LEAD utiliza ponderaciones, lógica de primer orden, y otras técnicas de inteligencia artificial como razonamiento basado en precedentes y aprendizaje

automático por refuerzo. LEAD fue implementado para trabajar con un subconjunto de figuras penales, sin embargo tiene una arquitectura abierta que facilita futuras extensiones sobre delitos, circunstancias y otras características propias.

LEAD demuestra que los sistemas informáticos pueden ser utilizados en el ámbito del Derecho para asistir en la toma de decisiones. LEAD es capaz de asistir a los operadores judiciales en el proceso de individualización de la pena, logrando que la misma sea predecible consistente y fundada en base a las pautas de valoración que fija el Código Penal Argentino.

LEAD utiliza esquemas de representación basados en frames (marcos), para la formalización de los conocimientos e implementación, sirve para especificar todas las características relevantes de los delitos y las circunstancias objetivas y subjetivas.

Una capacidad similar a la de HYPO de generar múltiples argumentos ha sido introducida a LEAD a través del razonamiento basado en precedentes.

Para la construcción de LEAD se uso como base los sistemas SHYSTER E HYPO, sacando lo mejor de ellos para su construcción.

1.4.2 Sistemas expertos legislativos

Los sistemas expertos han influido en tres aspectos básicos de la legislación, siendo estos:

1.4.2.1 Redacción de textos normativos

Uno de los aspectos más interesantes del desarrollo de los Sistemas Expertos reside en la posibilidad de una redacción automática de textos normativos, con lo que se lograría un gran avance en el desarrollo de la unificación de criterios de técnica legislativa para la redacción de textos normativos. Un texto redactado en lenguaje natural se podría convertir mediante estos Sistemas Expertos en otro texto más esquemático y riguroso. Algunos términos serían sustituidos por conectores de la lógica proposicional o por operadores deónticos.

1.4.2.2 Planificación del sistema legislativo en su conjunto

Permite evaluar el impacto de nuevas normas en el sistema jurídico y en el sistema social.

1.4.2.3 Proceso legislativo

Puesto que, una vez formada una base de conocimientos de legislación, el Sistema Experto jurídico es capaz de detectar las antinomias (contradicciones entre 2 o mas leyes), reiteraciones y lagunas existentes en los futuros proyectos de ley. Es en este punto donde más adelante se detendra para realizar un análisis más profundo dado que es la parte que se aplica en este trabajo de graduación.

1.5 Representación del conocimiento mediante marcos

Ideado por Marvin Minsky, el concepto de Marco es muy parecido a la noción de un objeto en Programación Orientada a Objetos. Un marco es una colección de atributos o slots, que poseen determinados valores y que

describen una entidad del mundo. Los marcos se agrupan en sistemas de marcos, relacionados unos con otros, modelando así las relaciones que existen entre las entidades del mundo representado. Además ofrece una estructura mucho más compacta que la lógica de predicados para representar conceptos y relaciones entre conceptos.

Los marcos nacidos en el mundo de la inteligencia artificial, son el origen de la orientación a objetos. De hecho, marco y objeto son prácticamente sinónimos.

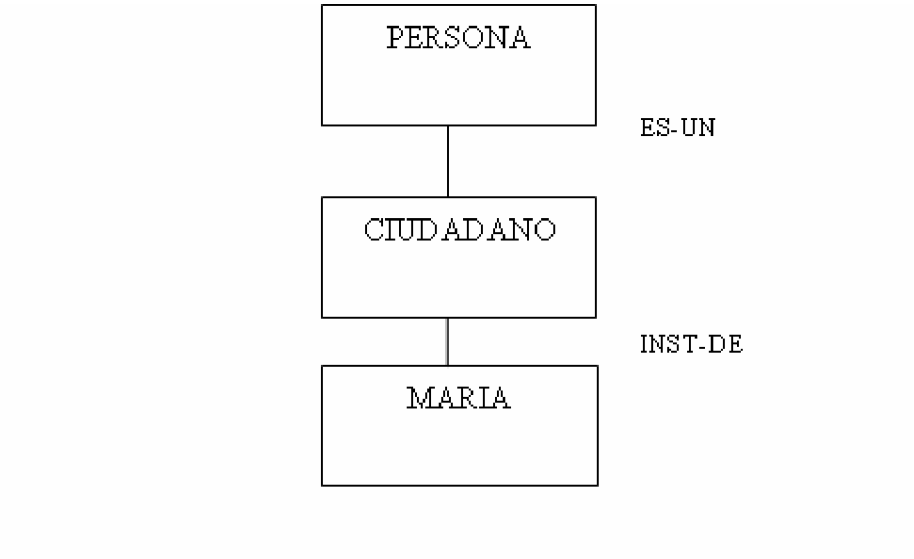
1.5.1 Propiedades de los marcos

De todas las relaciones existentes entre entidades del mundo, son especialmente relevantes para los sistemas de marcos las relaciones que expresan la pertenencia de un elemento a un conjunto (relación INSTANCIA-DE) y las que expresan la relación de subconjunto (relación ES-UN). Esto conduce a una organización jerárquica de los marcos, según clases, subclases y elementos.

1.5.1.1 Herencia de Propiedades

Las propiedades o slots de los marcos situados más arriba, y que corresponden por tanto a los conceptos más generales, se transmiten a los marcos que representan clases de objetos situadas más abajo en la jerarquía, así como a las instancias de dichas clases. Dicha transmisión de propiedades de un conjunto a sus subconjuntos, se denomina herencia.

Figura 1. Herencia de las propiedades



Así, por ejemplo, al ser CIUDADANO subconjunto de la clase PERSONA, heredará las propiedades de Sexo, edad, la instancia MARIA, hereda los slots correspondientes a CIUDADANO, y por tanto, también los de PERSONA.

1.5.1.2 Valores por omisión y excepciones

Habitualmente, en muchos dominios de la vida real aparecen excepciones, sobradamente conocido es el caso de que todas las aves pueden volar, pero no los pingüinos, o que ningún mamífero puede volar, salvo el murciélago. Los sistemas de marcos no tienen ningún problema en tratar correctamente estos casos.

1.5.1.3 Referencias a otros marcos

No sólo se pueden relacionar unos marcos con otros a través de las relaciones ES-UN e INSTANCIA-DE. En muchos casos, los valores de slot de un marco pueden apuntar a otro marco, para expresar un hecho que implica a ambos.

1.5.1.4 Jerarquía e inferencia

La jerarquía que se maneja en los marcos es parecida a la de un árbol. Y la forma de realizar la equiparación es por medio de reglas, por gestión dinámica de los valores, o por herencia siendo esta simple o compuesta.

1.5.1.4.1 Valores dinámicos

Al momento de añadir o modificar un valor de una propiedad y esta tenga un valor asociado, deduce el valor de la propiedad asociada.

1.5.1.4.2 Herencia múltiple

Un nodo puede tener un número arbitrario de superclases que lo contienen, permitiendo que un nodo herede las propiedades de múltiples nodos progenitores y de sus antecesores en la red.

A continuación se detallan 2 ejemplos, el primero explica cuándo un marco hereda a otro, el segundo en cambio no es posible que herede, por las condiciones propias que se dan:

- Si X<A<B y tanto A como B tienen la propiedad P, entonces X hereda la propiedad de A.
- Si X<A y X<B pero ni A<B ni B<Z y además si tanto A como B tienen la propiedad P con diferentes valores inconsistentes entre sí, entonces X no heredará nada ni de A ni de B.

1.5.2 Ventajas

Los marcos, como forma de representación del conocimiento ofrecen las siguientes ventajas:

- Facilita el procesamiento de la información
- Organización del conocimiento
- Capacidad de almacenar valores dinámicos
- Poder de abstracción.
- Encapsulamiento o capacidad de esconder información.
- Herencia, es decir pueden recibir características o propiedades de sus ancestros.
- Polimorfismo, que permite crear una interfaz común para todos los diversos objetos utilizados dentro del dominio.
- Posibilidad de reutilización del código.
- Mayor facilidad para poder trabajar eficientemente con sistemas grandes.

2 ANÁLISIS DEL SISTEMA EXPERTO LEGISLATIVO

En este capítulo se describe el análisis del sistema experto construido, a través del cual es definido el problema y se obtiene así una visión global de lo que se espera que realice el mismo, a la vez se define un campo de delimitación de las tareas que éste debe realizar, también se puede observar la metodología a utilizarse para resolver el problema que es de interés. A continuación se definirá el problema y las demás tareas que conciernen al sistema.

2.1 Definición del problema

El Prototipo de Sistema Experto realizado para el actual trabajo de Graduación, realiza lo que comúnmente en Derecho se llama “detectar antinomias”, es decir, contradicción entre dos leyes. El fundamento o base en este trabajo de graduación es la Constitución de la República de Guatemala y la iniciativa de ley que debe ser comparada con la misma.

El sistema experto emite un juicio con respecto a la iniciativa de ley, dictándola constitucional o inconstitucional según sea el caso, fundamentando su dictamen en base a los artículos, que fueron tomados en cuenta por el sistema para emitir el juicio correspondiente.

2.2 Delimitación del campo del sistema experto

Existen diferentes leyes, normas o reglas que regulan el comportamiento, derechos y obligaciones de los habitantes de una nación. En este trabajo de

graduación el punto central es la Nación de Guatemala y las leyes que rigen a la misma, específicamente la Constitución de la República de Guatemala de 1985, la cual es la suprema ley que regula a las demás leyes guatemaltecas. Es por eso que el sistema experto se clasifica como un sistema experto legislativo, porque el campo de acción del sistema experto serán algunas de las leyes comprendidas en la constitución de la república que son evaluadas por el legislador.

2.3 Tareas que realiza el sistema experto

Determinación de constitucionalidad en una propuesta de ley, es decir, si una propuesta de ley no contradice ninguno de los artículos comprendidos en la constitución de la república de Guatemala, esta propuesta de ley es constitucional.

Determinación de inconstitucionalidad en una propuesta de ley, se dice que una propuesta de ley es inconstitucional si en algunas de sus premisas contradice lo establecido en alguno de los artículos de la constitución de la república y en base a que artículos de la constitución de la república de Guatemala se baso el sistema para emitir tal dictamen.

2.4 Dominio del conocimiento

El conocimiento de la constitución de la república a tomarse en cuenta, abarcara conceptos tales como: la persona humana, derechos de la misma, fines y deberes del estado.

Dado que la constitución de la república es extensa y los campos que abarca son diversos, el dominio de conocimiento del sistema experto serán los

artículos del 1 al 10 de la constitución de la república de Guatemala a excepción de su artículo 5.

2.5 Utilización de marcos en la representación del conocimiento legislativo

En terminología jurídica un concepto esta enlazado con otro, por ej.: un ser humano puede ser un niño, un anciano, un indígena, etc. y un ciudadano es un ser humano, aunque no cualquier ser humano es un ciudadano (un ciudadano es una persona mayor de 18 años, con derecho a elegir y ser electo).

Como se puede observar la representación del conocimiento es abstracta, los marcos no solo se prestan para representar el conocimiento sino también para extraer el conocimiento, lo cual resuelve el problema de abstracción que representan los conceptos del conocimiento.

En los marcos se maneja una estructura conceptual, lo cual es de suma importancia en este estudio, dado que un concepto necesita que se incluya entre sus partes al concepto subordinado y que actúe de superordinado, esto permite armar una red de relaciones entre los conceptos y soluciona el problema estudiado en el capítulo 1, el concepto subordinado es ciudadano y el superordinado es ser humano.

Otra ventaja de realizar el Sistema Experto con marcos, es que al organizar los conceptos permite inferir en base a los atributos de aquellos conceptos que se encuentren en una jerarquía de orden superior, sin tener que estar especificando uno por uno.

Se observa que con esta forma de representación se crea una base de conocimiento de forma organizada, según una estructura compleja, ya que la representación no incluye sólo datos sino posibles reglas para que el sistema experto pueda ejecutarlas.

Existen varias ventajas en la utilización de una base de Conocimientos representada en Marcos, por ejemplo la herencia de valores por defecto, permite ahorrar tiempo y trabajo, otra ventaja es que el sistema experto puede deducir y complementar las propiedades de un concepto basándose en las propiedades de conceptos más altos en la jerarquía, además los valores de los conceptos son dinámicos.

En conclusión la metodología a utilizar para el presente trabajo de graduación es Marcos, por las características que mejor se adaptan al Sistema Experto que se desea construir.

2.6 Proceso de inferencia

Las conclusiones a las que se ha llegado al analizar al legislador en su campo de estudio, es que la lógica que utiliza proviene de diferentes fuentes, como por ejemplo: basado en casos, estudios, relaciones con otras leyes, etc., y cada persona tiene su propia interpretación de la ley. Llegar a este punto llevaría un estudio sumamente extenso, en base a lo anterior el prototipo de sistema experto que se ha realizado tiene 2 fases

- Encontrar reiteraciones, es decir, si para la iniciativa de ley existe ya alguna ley parecida, entonces indica el articulo existente, para darle una pauta al legislador en la toma de decisiones.
- Encontrar antinomias, es decir contradicción entre dos leyes, de tal manera que no contradiga explícitamente la ley o bien alguna parte

del dominio, por ejemplo “El estado no garantiza la vida del embrión”, este proyecto de ley contradice lo que dice en las constitución en el artículo 3, en donde dice que el estado garantiza la vida del ser humano y un embrión es un ser humano.

3 BASE DE CONOCIMIENTOS LEGISLTATIVOS, OBTENCION Y REPRESENTACION

3.1 Introducción

Ahora que se ha definido el problema y se sabe cual es el punto al que se desea llegar, se enfrenta con el problema de obtener y representar el conocimiento de la Constitución de la República de Guatemala, en el campo delimitado anteriormente.

Lo que se realiza en este capitulo es obtener todos los conceptos de los artículos del 1-10 de la constitución de la República de Guatemala y también los atributos de los mismos, que servirán para construir el sistema experto legislativo.

A continuación se vera cual es el papel que juegan los conceptos con la resolución del problema y proseguido a este se exponen todos los conceptos con las relaciones que existen entre los mismos.

3.2 Conceptos doctrinarios y legales (Obtención del conocimiento)

A continuación se vera la relación que tienen los conceptos con la representación del conocimiento, por ejemplo ¿quiénes son personas?, y para esto es necesario obtener el concepto legal de quienes son declaradas personas, luego de tener el concepto podemos saber que una persona tiene un nombre, una dirección, un estado civil, etc., también podemos saber que una persona es:

Según el Sexo:

- Mujer
- Hombre

Según la edad

- Niño
- Menor de Edad
- Mayor de Edad
- Anciano
- Adolescente, etc.

Según el estatus social

- Pobre
- Rico

Según la profesión

- Doctor
- Enfermera
- Licenciado
- Albañil, etc.

Todos los anteriormente mencionados son personas y deberían estar incluidas en el marco persona de una forma directa o heredada, cualquiera sea el caso. Pero como todos estos hechos no los podemos saber ahora, o pueden incrementarse en el futuro, es necesario dejar bien definidos los marcos y una estructura jerárquica para los mismos, para que en el futuro el usuario tenga la oportunidad de agregar nuevo conocimiento (hechos), para los conceptos que se definirán ahora por medio de representación por marcos.

La siguiente tarea es obtener los conceptos doctrinarios y legales que respalden la estructura de los marcos, dichos conceptos son enunciados a continuación.

Persona

Desde el ángulo del Derecho actual, la persona es el sujeto de derecho, o, por mejor decir, el ser susceptible de tenerlos o de figurar como término subjetivo en una relación de derecho. Ahora bien, si esta ambivalencia es exacta desde un punto de vista abstracto, sufre reparos si la frase "sujeto de derecho" se toma en una acepción concreta, significando a quien está investido actualmente de un derecho determinado. En tales respectos podemos decir que el término "persona" es más amplio que el desujeto de derecho. Así, un niño y un loco serán personas, pero existen serios obstáculos para considerarlos como sujetos de derecho en los términos técnicos de la ciencia jurídica. Todo sujeto de derecho, pues, será persona; pero no toda persona será sujeto de derecho, porque la actuación concreta supone aptitud o susceptibilidad, pero no viceversa. (13 -235,236)

Estado Civil

Conjunto de cualidades inherentes a la persona, tomada en consideración por la ley civil para asignarle determinados efectos. Condición del individuo dentro del orden jurídico, que influye en sus facultades, capacidad y obligaciones. Así, como factores del estado civil: la calidad de nacional o extranjero; la edad; la condición de casado o soltero; la de hijo o padre; el sexo, etc. (4-19)

Estado

El Estado es una sociedad humana establecida en el territorio que le corresponde, estructurada y regida por un orden jurídico, que es creado,

definido y aplicado por un poder soberano, para obtener el bien público temporal, formando una institución con personalidad moral y jurídica. (12 - 26,27)

El Estado es una organización social constituida en un territorio propio, con fuerza para mantenerse en él e imponer dentro de él un poder supremo de ordenación y de imperio, poder ejercido por aquel elemento social que en cada momento asume la mayor fuerza política. (10 -294)

Derecho

Sistema coactivo de normas generadoras de autorizaciones y deberes, que tiene por objeto ordenar de cierto modo la conducta de los hombres dentro de las relaciones sociales que establecen, tendientes a la satisfacción de sus necesidades en una organización estatal determinada, con el fin de mantener dicha organización y lograr la realización de los intereses a ella inherentes. (4 - 22)

Facultad de hacer una cosa, de disponer de ella o de exigir algo de la persona.

Libertad

Estado existencial del hombre en el cual éste es dueño de sus actos y puede autodeterminarse consciente sin sujeción a ninguna fuerza o coacción psicofísica interior o exterior.

La libertad representa un concepto contrario al determinismo y ofrece extraordinaria importancia en relación al Derecho Político; ya que la libertad es

el fundamento no ya de un determinado sistema de vida, sino de la organización del Estado.

Éste es un concepto de una amplitud ilimitada, que obedece además a un dilatado itinerario histórico. La libertad era, entre los antiguos griegos, el don o la facultad del hombre libre, es decir, el no esclavizado, el no sometido, para actuar según su voluntad. La idea de libertad envolvía no solamente el poder de escoger y decidir sino también el de autodeterminarse. La noción de la libertad incluía, por tanto, la de responsabilidad para consigo mismo y para con la comunidad.

Este concepto se consagró en la declaración de los Derechos del Hombre y del Ciudadano, aprobada en Francia el 26 de agosto de 1789: “La libertad consiste en poder hacer todo lo que no dañe a otro”, de modo que “el ejercicio del derecho natural de cada hombre no tiene más limites que aquellos que aseguran a los demás límites que aquellos que aseguran a los demás miembros de la sociedad el goce de los mismos derechos”. (3-614,615)

Justicia

Este término, que viene del latín iustitia, tiene muchas significaciones. En el sentido político de la palabra, justicia es lo que, con referencia a un todo, corresponde a cada quien en derecho y deberes frente al Estado y la sociedad. En sentido administrativo, es el tribunal o judicatura que oye y juzga a las partes de un litigio.

Virtud que inclina a dar a cada uno lo que le corresponde. En sentido jurídico equivale a lo que es conforme al Derecho.

En otro sentido, se entiende por justicia la organización judicial de un país; y así se habla de tribunales de justicia, Corte Suprema de Justicia, justicia civil, justicia penal, justicia administrativa, justicia militar. (10 -411)

Seguridad

Exención de peligro o daño. Solidez. Certeza plena. Firme convicción. Confianza. Fianza. Garantía. Ofrecimiento de cumplir o hacer para determinado plazo. Sistema de prevención racional y adecuada. (10 -695).

Paz

La paz es el propósito declarado de todas las religiones, convicciones filosóficas, ideológicas políticas, naciones, etnias y credos. (3-771)

Desarrollo Integral

Crecer, desenvolver, crecer, aumentar, acrecentar, perfeccionar, mejorar.

Integridad

Calidad de íntegro, entereza, desinterés: la calidad de un magistrado. Virginidad. (13).

Igualdad

Del concepto genérico, como conformidad de una cosa con otra en naturaleza, calidad o cantidad, se desprenden diversas consecuencias que pueden afectar al orden jurídico. La primera de ellas tiene su origen en la

determinación de si la idea de igualdad representa una realidad o una mera teoría. No puede llegarse a una conclusión sin distinguir entre el hombre considerado en sus condiciones naturales, como criatura humana, y el hombre en relación a sus características, como integrante de una sociedad organizada. En el primer sentido no puede decirse que exista igualdad, aun cuando se dé semejanza, porque no todas las personas tienen el mismo grado de inteligencia, de fortaleza, de belleza, de iniciativa, de valor. De esas diferencias se deriva una consideración distinta de los hombres frente a la ley, debiéndose tomar esta afirmación en el sentido de que, mientras unos tienen plena capacidad para gobernar sus actos por sí mismos; otros, en razón de la edad, de la deficiencia mental o de la enfermedad y hasta, en ocasiones, del sexo, no tienen capacidad para actuar jurídicamente o la tiene disminuida. Inclusive frente a un mismo hecho delictivo, esa misma diferencia de condiciones personales puede llevar desde la plena imputabilidad del acto hasta la absoluta inimputabilidad.

Por eso, cuando en términos de Derecho se habla de igualdad, lo que se quiere decir es que la ley no establece distinciones individuales respecto a aquellas personas de similares características; ya que a todas ellas se les reconocen los mismos derechos y las mismas posibilidades. (10 -362).

Ser Humano

Un ser humano existe desde su concepción hasta su muerte.

Servidumbre

Estado o condición de siervo. Sujeción grave u obligación inexcusable de hacer algo. (7)

Ley

Constituye la ley una de las fuentes, tal vez la principal, del Derecho. En sentido amplio, se entiende por ley toda norma jurídica reguladora de los actos y de las relaciones humanas, aplicable en determinado tiempo y lugar. Dentro de esa idea, sería ley todo precepto dictado por autoridad competente, mandando o prohibiendo una cosa en consonancia con la justicia y para el bien de los gobernados. Así, entrarían dentro del concepto no sólo la ley en sentido restringido o propio, como norma jurídica elaborada por los órganos estatales con potestad legislativa.

La ley, en la moderna teoría general del Derecho, puede se tomada en dos aspectos uno formal, que se refiere a la que ha sido dictada por el Poder Legislativo conforme a los procedimientos específicamente preestablecidos; y otro material, que alude a toda norma jurídica cuyo contenido regula una multiplicidad de casos, haya sido dictada o no por el órgano legislativo. Esta división coincide con la antes expuesta sobre el concepto amplio y estricto de la ley. (10 -424).

Delito

Son varias las definiciones que en la doctrina y en algunos códigos penales se han dado al delito. Recogiendo la de Jiménez de Asúa, se entiende por tal “el acto típicamente antijurídico, culpable, sometido a veces a condiciones objetivas de penalidad, imputable a un hombre y sometido a una sanción penal”. En consecuencia, según ese mismo autor, las características del delito serían: actividad, adecuación típica, antijuricidad, imputabilidad, culpabilidad, penalidad y, en ciertos casos, condición objetiva de punibilidad.

Soler lo define como “una acción típicamente antijurídica, culpable y adecuada a un figura legal conforme a las condiciones objetivas de ésta”; por lo cual sus elementos sustantivos son; la acción, la antijuricidad, la culpabilidad y la adecuación a una figura. Para la definición de Carrera, en la cita de Soler, es la “Infracción de la ley del Estado, promulgada para seguridad de los ciudadanos, resultante de un acto externo del hombre, positivo o negativo, moralmente imputable y políticamente dañoso”. (10 - 212).

Nuestro Código penal no define el delito. La comisión redactora supuso, como lo señalan connotados tratadistas de derecho penal, que es difícil asentar una buena definición. Nosotros, sin embargo, si estamos con la idea de que el Derecho Penal se contrae al contenido del los Códigos respectivos, nos aventuramos a decir, siguiendo a Gerland, que por delito debe entenderse una acción o una omisión antijurídica y culpable que señala la ley penal. Casi a semejanza de las definiciones que traen algunos Códigos Latinoamericanos y el de España. Tal definición comprende la tesis de que el derecho penal descansa en el binomio: delito y pena.

El comportamiento delictivo puede manifestarse por actividades positivas (de acción) o negativas (de omisión), revestidas de antijuridicidad.

Para Liszt, en sentido formal: El hecho al que el ordenamiento jurídico atribuye, como consecuencia jurídica, una pena; y, en sentido material, el acto culpable, contrario al derecho y sancionado con una pena.

Nos gusta, por su sentido moderno, la definición de Carlos Sotos, dice que el delito es el daño o riesgo culpable de un bien que el legislador penal debe proteger, en nombre del interés público, contra cualquier ataque mediante la eficacia de la pena. (7 -21, 22)

Falta

Esta voz, que tiene muchas acepciones gramaticales, es también susceptible de diversas interpretaciones jurídicas, la más caracterizada de las cuales tal vez sea la que afecta a su sentido penalístico; ya que se entiende por tal, según la definición de la Academia, la "infracción voluntaria de la ley, ordenanza, reglamento o bando, a la cual esta señalada sanción leve".

El concepto incurre en un error, porque la infracción puede ser, y corrientemente es, originada no por dolo (que sería la característica de la voluntariedad), sino por simple culpa derivada de imprudencia o negligencia, pero ya con una calificación: la de falta de intención. (10 - 312).

Autoridad Judicial Competente

Es la jurisdicción que ejerce un juez de conformidad con la materia o territorio asignado por la Corte Suprema de Justicia.

Notificación

Acción y efecto de hacer saber a un litigante o parte interesada en un juicio, cualquiera sea su índole, o a sus representantes y defensores, una resolución judicial u otro acto del procedimiento. Couture dice que es también constancia escrita, puesta en los autos, de haberse hecho saber a los litigantes una resolución del juez u otro acto del procedimiento. (10 -489).

Detención

Privación de la libertad de quien se sospecha autor de un delito; tiene carácter preventivo y previo a la presentación del mismo ante el juez. (10 - 250).

Preso

Persona detenida por sospechosa y contra la cual se ha dictado auto de prisión preventiva, que obliga a permanecer en establecimiento carcelario. La situación es revocable hasta verse el proceso. Condenado por sentencia a una pena privativa de libertad que cumple en local penitenciario. (10 - 603).

Documentación

Probanza o justificación de una cosa, mediante escritos. Conjunto de documentos que para tales fines se emplea. Documentos de identidad. Serie de antecedentes, certificaciones, partidas, autorizaciones, exigidos para determinados trámites o solemnidades, ya sea para el matrimonio, para lograr un pasaporte, para la exportación, entre tantos casos en la desbordada burocracia de hoy. (10 -254).

Sanción

En el orden penal es el castigo que el juez impone a un delincuente o infractor de la ley.

La sanción es la pena o castigo que la ley prevé para su aplicación a quienes incurran o hayan incurrido en una infracción punible. (10 - 688, 689).

Condena

Decisión judicial por la cual se obliga a una de las partes en juicio a satisfacer las pretensiones de la otra, ya sea en todo o en parte, Según Couture, es la "determinación judicial de la conducta debida por un litigante, al que se impone la obligación de dar, hacer u omitir algo, bajo amenaza implícita y eventual de coacción". En materia penal, decisión judicial represiva que individualiza una pena contra el autor de una infracción o delito. (10 -146).

Pena

Castigo impuesto por autoridad legítima, especialmente de índole judicial a quien ha cometido un delito o falta. Mezger dice que en sentido estricto es "la imposición de un mal proporcionado al hecho"; es decir, una "retribución" por el mal que ha sido cometido. Y en sentido auténtico, la pena es la que "corresponde, aun en lo que respecta al contenido, al hecho punible cometido", debiendo existir entre la pena y el hecho "una equiparación valorativa" (10 - 558).

Auto

Resolución judicial por la que se deciden cuestiones de importancia afectantes a intereses de los litigantes dignos de protección pero distintos de la cuestión de fondo, esto es, del objeto principal y necesario del proceso. Así, mediante auto se suelen resolver cuestiones como las incidencias, los relativos a presupuesto procesales, los recursos contra providencias, etc. Los autos se formulan expresando no sólo el tribunal que los dicta y su contenido, sino también su motivación, mediante la exposición en párrafos separados y numerados de los antecedentes de hecho y los fundamentos de derecho.

Prisión

Establecimiento carcelario donde se encuentran los privados de libertad por disposición gubernativa o judicial. Nombre de una pena privativa de libertad, de duración y carácter variables de un país a otro. (10 - 609).

Arresto

Detención provisional del presunto reo. Reclusión por tiempo breve como corrección o pena. En la escala de penas del Código Penal español de 1870, reformado en 1932, se establecen la penas de arresto mayor, con duración de un mes y un día a seis meses, y la de arresto menor, con duración de uno a treinta días, además de sus accesorias. Similarmente, en el Código Penal italiano. Estas penas, por su levedad, se aplica a los delitos de escasa importancia y, principalmente, a las llamadas faltas o contravenciones,

Con referencia al Derecho Procesal, es el acto ejecutado por autoridad competente de aprehender a una persona de la que se sospeche haya cometido un delito o contravención, y retenerla detenida por breve tiempo, hasta que intervenga el juez que ha de entender en el asunto. En definitiva, el arresto equivale a lo que otras legislaciones, entre ellas la argentina denominan detención.

Multa

Pena pecuniaria que se impone por una falta, exceso o delito, o por contravenir a los que en con esta condición se ha pactado.

En el Derecho Penal constituye una de las sanciones más benignas que se imponen por la comisión de determinados delitos. Asimismo es frecuente la imposición de multas de orden administrativo, con respecto a la comisión de determinadas infracciones, sea de orden municipal o de carácter fiscal. Civilmente, las multas pueden imponerse como sanción por el incumplimiento de algunas obligaciones; pero en este caso más revisten el carácter de indemnización de perjuicios o de cláusulas penales establecidas en los contratos. (10 - 474).

3.3 Representación del conocimiento

En el inciso anterior se ha presentado el conocimiento obtenido, ahora es tiempo de representar el conocimiento, este se mostrara de la forma más simple, explicando los atributos de los conceptos obtenidos del inciso anterior.

Persona

- Nombre: es la palabra o signo de individualización que se inscribe en el registro civil y sirve para distinguir al hombre de los demás, constituye el principal elemento de identificación de las personas. Es un medio para designar a las personas y constituye un derecho

subjetivo intelectual y carácter eminentemente extramatrimonial. (6 - 24)

- Estado Civil, ref. marco Estado Civil.
- Domicilio: se trata del asiento legal de una persona, el lugar donde se le considera establecida para el cumplimiento de sus obligaciones y el ejercicio de sus derechos subjetivos. (6 - 28)

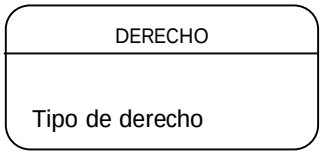
Estado Civil

- Matrimonio, pudiendo ser
  - Casado
  - Soltero
- Nacionalidad
  - Nacional
  - Extranjero
- Edad
  - Menor de edad
  - Mayor de edad

Estado

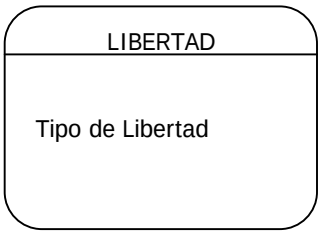
- Territorio: es el área geográfica dentro de la cual el Estado ejerce su poder, su soberanía.
- Población: es el conglomerado humano compuesto por las personas individuales que han nacido dentro del territorio del Estado y extranjeras, que por razón de domicilio vivan permanentemente dentro de el.
- Organización política: conjunto de mecanismos que permiten al grupo dirigente, monopolizar el poder decisorio para controlar y garantizar la vida y organización del propio Estado, sus instituciones y hacer cumplir sus decisiones que por devenir de esa fuente, tienen el sello legal.
- Orden jurídico: conjunto de normas positivas vigentes relacionadas entre sí y escalonadas o jerarquizadas, que rigen en cada momento la vida y las instituciones de todas clases dentro de una nación determinada.

Derecho



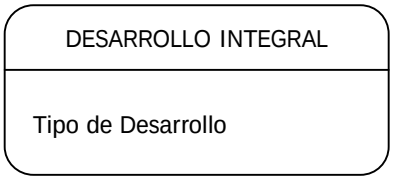
- Tipo de derecho: siendo vida, libertad, seguridad, educación.

Libertad



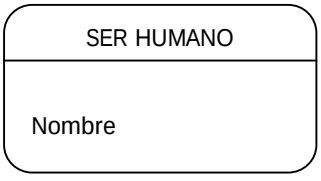
- Tipo de libertad: pudiendo ser, libertad civil, derecho de hacer todo cuanto no prohíbe la ley; libertad de conciencia, derecho de profesar cualquier opinión religiosa; libertad de comercio, facultad de comprar y vender sin estorbo alguno; libertad condicional, la que se concede al penado bajo ciertas condiciones; etc.

Desarrollo Integral



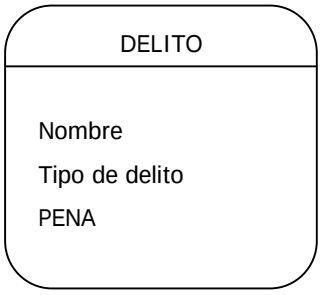
- Tipo de desarrollo, ejemplos de estos es educación, salud, etc.,

Ser Humano



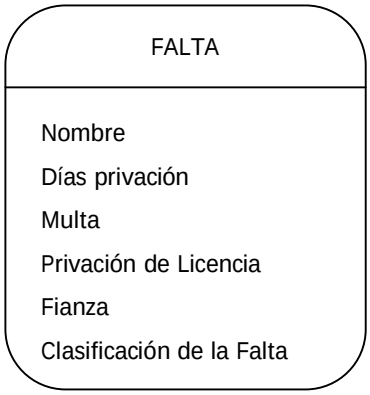
- Nombre: especifica el nombre que se le da al ser humano.

Delito



- Nombre
- Tipo de delito, pudiendo ser un delito fiscal, delitos contra el honor, etc.,
- PENA, en donde se especifica la sanción que se aplica al delito.

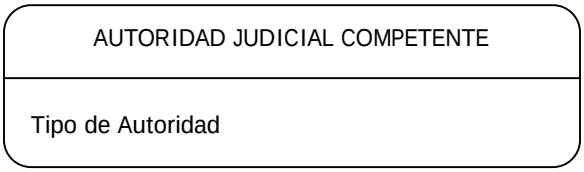
Falta



- Nombre: nombre que recibe la falta.
- Días privación: que comprende de uno a sesenta días de privación de libertad.
- Multa: sanción pecuniaria que recibe un sindicado de cometer una falta de conformidad con la ley.
- Privación de licencia: privación de la licencia de conducir.

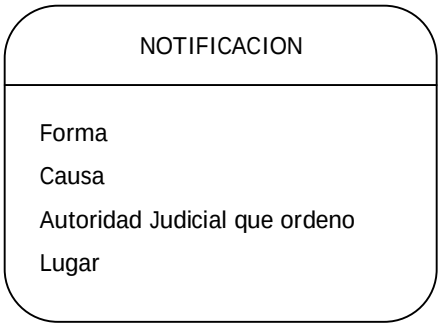
- Fianza: medida sustitutiva que recibe un sindicado para obtener su libertad preventiva mientras se continúa con la investigación.
- Clasificación de la falta: clasificación de la misma

Autoridad Judicial Competente



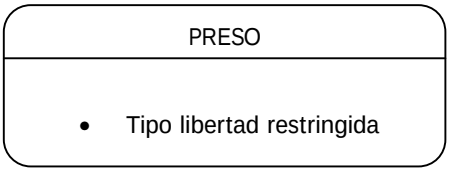
- Tipo de autoridad: especifica al titulo que se le da a esta autoridad, como ejemplo: juez, magistrados, corte de suprema justicia, etc.,

Notificación



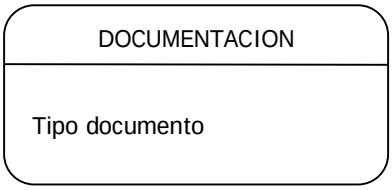
- Forma: la forma en que se notifico, debiendo ser verbal y por escrito, si la notificación no lleva ambos es invalida.
- Causa: causa o motivo de la razón por la que se le detiene.
- Autoridad judicial: la autoridad que ordeno la resolución, debiendo ser, una autoridad competente.
- Lugar: el lugar en el que permanecerá.

Preso



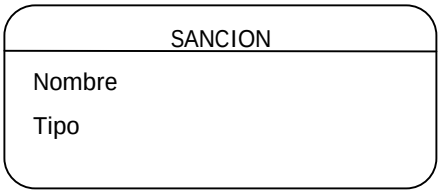
- Siendo tipo libertad restringida, el tipo de libertad que se le restringe a la persona presa.

Documentación



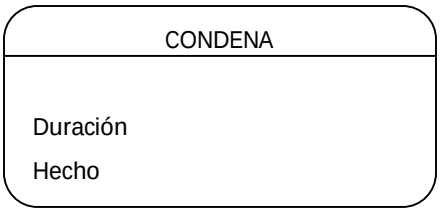
- Tipo de documento: nombre que recibe el documento, pudiendo ser, cédula, licencia, fe de edad, etc.,

Sanción



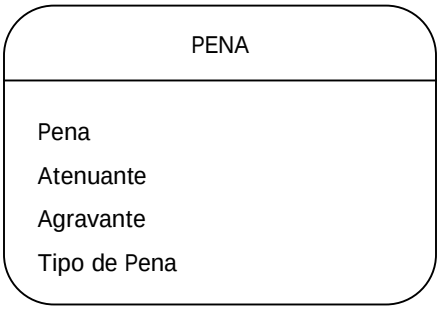
- Sanción: el nombre de la sanción
- Tipo: se especifica el tipo de sanción.

Condena



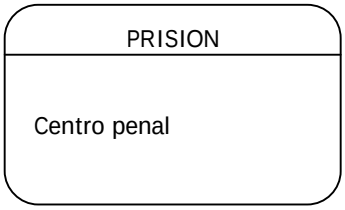
- Duración, tiempo que se ha establecido para cumplir la condena.
- Hecho, es el motivo por el que se impone la condena.

Pena



- Pena, nombre del tipo de pena pudiendo ser esta la pena de muerte, la pena de prisión, la pena de arresto, la pena de multa, etc.,
- Tipo de Pena, pudiendo ser privativa de libertad, restrictiva de libertada, restrictiva de derechos, pecuniaria, etc.,
- Atenuante, situación que disminuye la gravedad de la pena.
- Agravante, situación que empeora o aumenta la situación de algo.

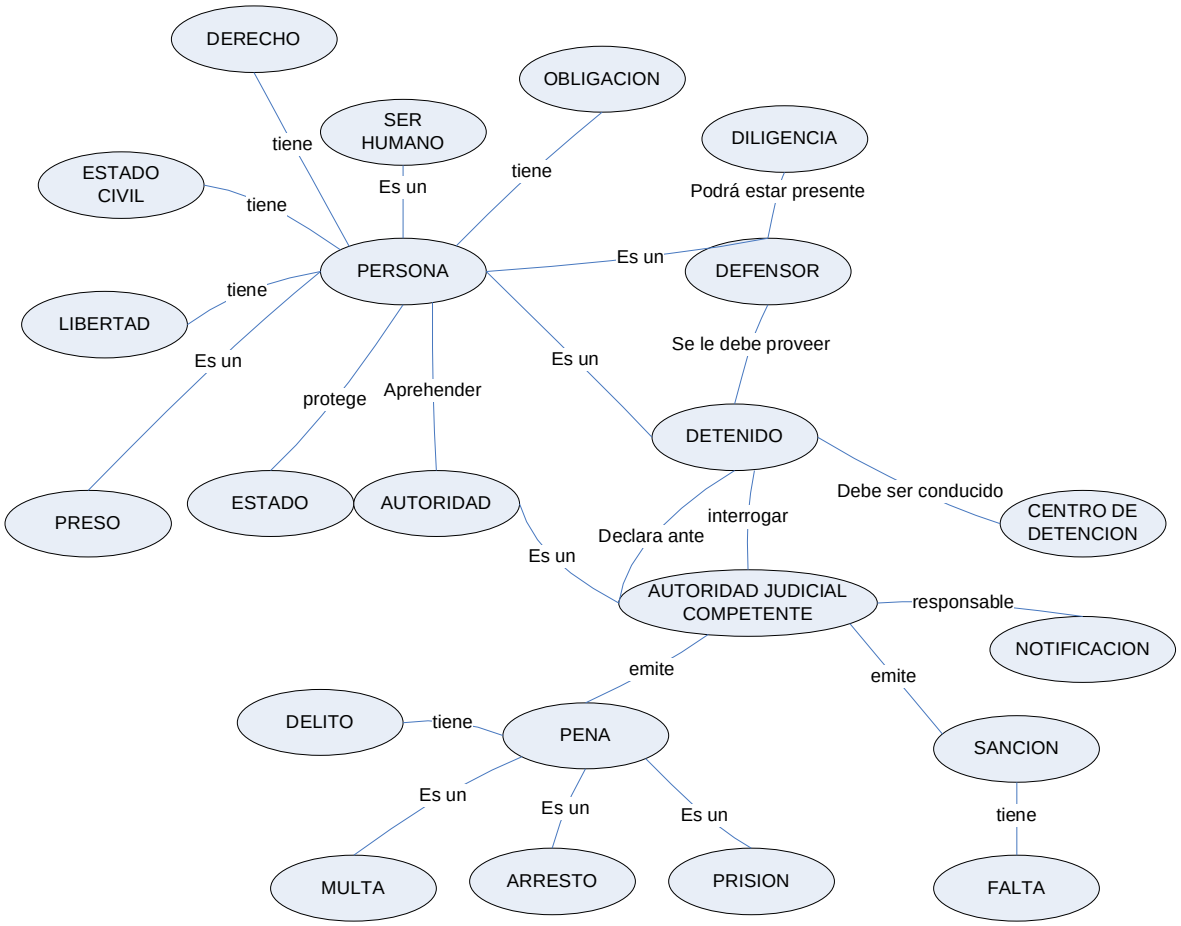
Prisión



- Centro Penal, es el lugar establecido para que se cumpla la prisión.

Habiendo obtenido y definido los conceptos legales que se abstraen en este prototipo, es necesario mostrar la relación que existe entre ellos. En la figura 2, que se muestra a continuación se detalla la relación que existe entre estos conceptos.

Figura 2. Relación de los conceptos doctrinarios



3.4 Definición de reglas

A continuación se definen las reglas que utiliza el Sistema Experto Legislativo, dándolas a conocer de una forma descriptiva para su mayor compresión, estas hacen también referencia al artículo del que fue tomado para su abstracción.

- El estado de Guatemala protege a toda Persona. Ref. Art. 1
- El estado de Guatemala garantiza la vida a los habitantes de la republica y la libertad a los habitantes de la republica y la seguridad a los habitantes de la republica y la paz a los habitantes de la republica y el desarrollo integral a los habitantes de la republica. Ref. Art. 2
- El estado de Guatemala garantiza y protege la vida del ser humano y la integridad de la persona y la seguridad de la persona. Ref. Art. 3
- Todos los seres humanos son libre e iguales. Ref. Art. 4
- Todas las personas tienen iguales oportunidades y responsabilidades.
- Ninguna persona puede ser sometida a servidumbre. Ref. Art. 4
- Ninguna persona puede ser detenida o presa, sino por causa de delito o falta y en virtud de orden librada con apego a la ley por autoridad judicial competente. Se exceptúan los casos de flagrante delito o falta. Ref. Art. 6
- Los detenidos deberán ser puestos a disposición de la autoridad judicial competente en un plazo no mayor de 6 horas. Ref. Art. 6
- Toda persona detenida deberá ser notificada inmediatamente, en forma verbal y por escrito, de la causa que motivó su detención, autoridad que la ordenó y lugar en el que permanecerá. Ref. Art. 7
- Todo detenido deberá ser informado inmediatamente de sus derechos en forma que le sean comprensibles. Ref. Art. 8
- Todo detenido deberá proveérsele de un defensor. Ref. Art. 6

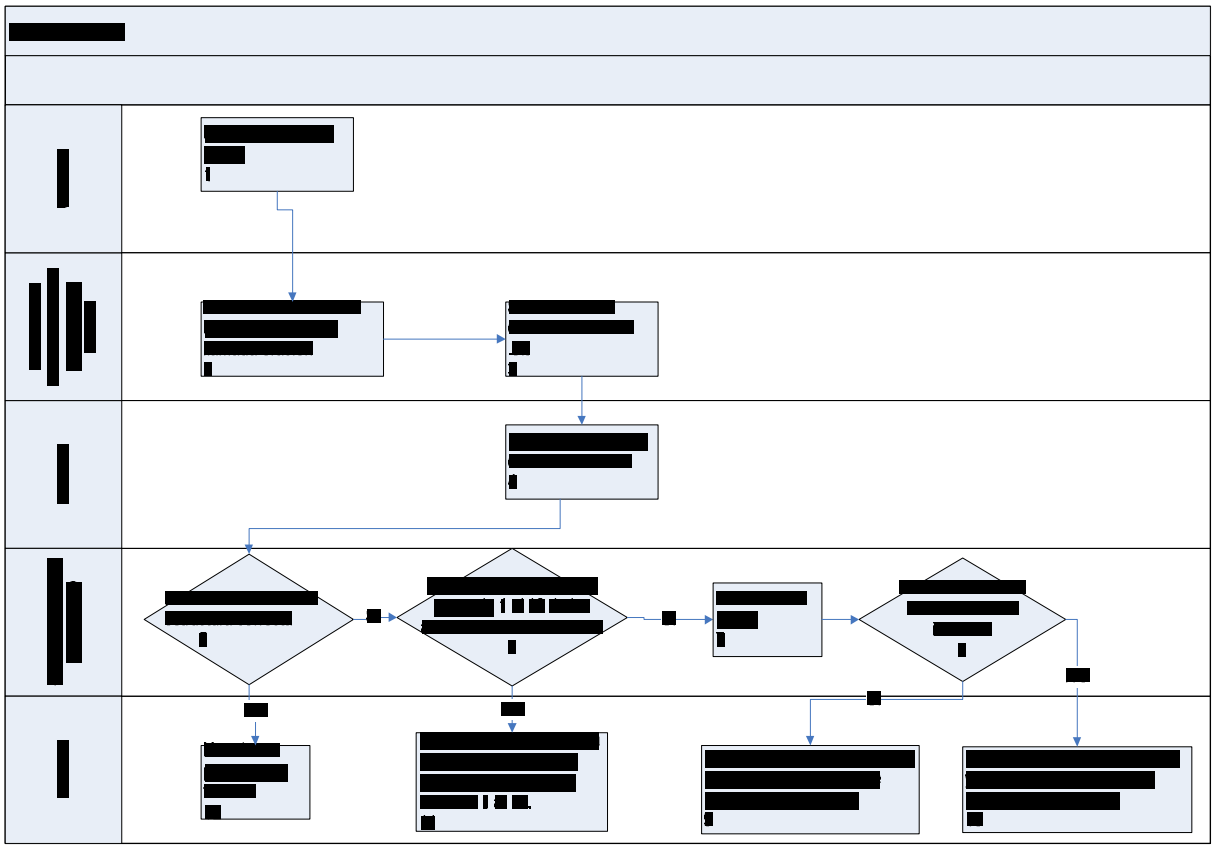
- Un defensor podrá estar presente en todas las diligencias policiales y judiciales del detenido. Ref. Art. 8
- El detenido no podrá ser obligado a declarar sino ante autoridad judicial competente. Ref. Art. 8
- Las autoridades judiciales son las únicas competentes para interrogar a los detenidos o presos. Ref. Art. 9
- La duración del interrogatorio a detenidos o presos no durara más de 24 horas. Ref. Art. 9
- No tiene valor probatorio el interrogatorio extrajudicial. Ref. Art. 9
- Las personas aprehendidas por la autoridad no podrán ser conducidas a lugares de detención, arresto o prisión diferentes a los que están legal y públicamente y públicamente destinados al efecto. Ref. Art. 10
- Los centros de detención, arresto o prisión provisional, serán distintos a aquellos en que han de cumplirse las condenas. Ref. Art. 10

4 DISEÑO DEL SISTEMA

4.1 Flujo del proceso

El Sistema Experto Legislativo, para su correcto funcionamiento, lleva un procedimiento y un ordenamiento lógico de las diferentes tareas que se llevan a cabo, desde el reconocimiento de lenguaje hasta el veredicto sugerido por el Sistema Experto Legislativo al usuario, a continuación se describe todo el flujo del Sistema para luego desglosarlo

Figura 3. Flujo del Proceso



En la figura 3, se detalla el flujo del sistema, tomándose en cuenta que del proceso 1 al 3 esta en proceso de construcción en otro trabajo de graduación, de los procesos 4 al 12 se trataran dentro de los subcapitulos que a continuación se describen.

4.2 Entradas

En lo que respecta a entradas o alimentación del Sistema Experto Legislativo para el reconocimiento de las iniciativas de ley, se vale de una estructura llamada "*oracion*", la cual el usuario debe ingresar, esta estructura almacena la información de la iniciativa de ley, en instancias de la misma.

4.2.1 Estructura de la oración

La estructura de cada oración ingresada por el usuario, entiéndase por oración la frase terminada con punto se guarda en instancias del objeto con el mismo nombre "oracion". A continuación se estudia la estructura de la misma y se brinda una descripción de sus atributos:

```
(defclass ORACION (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (multislot sustantivo_sujeto (create-accessor write))
  (slot verbo_aux (create-accessor write))
  (slot verbo_modal (create-accessor write))
  (multislot verbo (create-accessor write))
  (multislot adverbio (create-accessor write))
  (slot condicional (create-accessor write))
  (multislot sustantivo_predicado (create-accessor write))
  (slot neg_verbo (create-accessor write))
  (slot neg_sustantivo_sujeto (create-accessor write))
```

(slot neg_sustantivo_predicado (create-accessor write)))

- ***sustantivo_sujeto:*** en este multislot se almacenan los sustantivos del sujeto, por ejemplo: toda persona es libre en dignidad y derechos, lo cual dará como resultado el almacenamiento de la palabra “persona” en el multislot sustantivo_sujeto.
- ***verbo:*** en este multislot se almacenan los verbos, siguiendo con el ejemplo: toda persona es libre en dignidad y derechos, se obtendrá como resultado el almacenamiento del verbo “ser” en el multislot verbo, es almacenado el verbo en infinitivo a fin de que el Sistema Experto Legislativo y el Sistema de Reconocimiento de Lenguaje Natural, tengan un lenguaje en común.
- ***condicional:*** este slot almacena la condición encontrada en el predicado, en el caso de existir tal condición, de lo contrario almacenara un valor nulo. Si existiera una condición, como el siguiente ejemplo: Ninguna persona puede ser detenida o presa, sino por causa de delito o falta, dará como resultado el almacenamiento de la palabra “sino” en el slot condicional.
- ***sustantivo_predicado***: en este multislot se almacenan los sustantivos del predicado, por ejemplo: toda persona es libre en dignidad y derechos, lo cual dará como resultado el almacenamiento de las palabras “dignidad” y “derechos” en el multislot sustantivo_predicado.
- ***neg_sustantivo_sujeto:*** este slot almacena la negación encontrada en el sustantivo del sujeto, en el caso de existir tal negación, de lo contrario almacenara un valor nulo. Si existiera una negación, como el siguiente ejemplo: No toda persona es libre en dignidad y derechos, dará como resultado el almacenamiento de la palabra “no” en el slot neg_sustantivo_sujeto.

- ***neg_sustantivo_predicado:*** este slot almacena la negación encontrada en el sustantivo del predicado, en el caso de existir tal negación, de lo contrario almacenara un valor nulo.

4.3 Hechos

En esta sección se explica la estructura de las clases, estas clases son definidas en base a los conceptos explicados en el capítulo anterior, además se muestran algunas instancias que han sigo guardadas en la misma, esto con el fin de que se visualice que tipo de información almacena en las mismas, para que en un futuro se siga ampliando la base de conocimientos del Sistema Experto. Esta estructura esta definida en EHSIS, siendo EHSIS un Generador de Sistemas Expertos.

Como en la constitución todo gira alrededor de la persona, se comenzara viendo la estructura utilizada para almacenar las instancias de persona y la estructura que se utilizo para la misma.

- ***Persona:*** en esta clase se define la estructura que deben tener las instancias de persona, sabiendo que una persona es un *ser humano,* se dejará almacenado en persona el nombre de la instancia.

```
(defclass PERSONA (is-a SER_HUMANO)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias almacenadas en esta clase son:

```
(make-instance persona1 of PERSONA (nombre "hombre"))
(make-instance persona4 of PERSONA (nombre "habitante"))
(make-instance persona5 of PERSONA (nombre "mujer"))
```

(make-instance persona6 of PERSONA (nombre "varón"))

Las clases que se activaran cuando en una regla se hable de persona son las siguientes

- **MIEMBRO_FAMILIA:** las instancias de esta clase son los miembros de la familia, la estructura de esta clase es la siguiente:

  (defclass MIEMBRO_FAMILIA (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write)))

  Algunas de las instancias de la clase son las siguientes:

  (make-instance miembro1 of MIEMBRO_FAMILIA (nombre "tio"))
  (make-instance miembro2 of MIEMBRO_FAMILIA (nombre "abuelo"))
  (make-instance miembro3 of MIEMBRO_FAMILIA (nombre "hermano"))
  (make-instance miembro4 of MIEMBRO_FAMILIA (nombre "primo"))
  (make-instance miembro5 of MIEMBRO_FAMILIA (nombre "sobrino"))
  (make-instance miembro6 of MIEMBRO_FAMILIA (nombre "nieto"))
  (make-instance miembro7 of MIEMBRO_FAMILIA (nombre "bisnieto"))

- **PROFESIONAL:** las instancias de esta clase son los profesionales, la estructura de esta clase es la siguiente:

  (defclass PROFESIONAL (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write)))

  Algunas de las instancias de la clase son las siguientes:

  (make-instance profesional2 of PROFESIONAL (nombre "doctor"))
  (make-instance profesional3 of PROFESIONAL (nombre "licenciado"))

```
(make-instance profesional4 of PROFESIONAL (nombre "abogado"))
(make-instance profesional5 of PROFESIONAL (nombre "maestro"))
(make-instance profesional6 of PROFESIONAL (nombre "agricultor"))
(make-instance profesional7 of PROFESIONAL (nombre "ama de casa"))
(make-instance profesional8 of PROFESIONAL (nombre "tejedor"))
```

- **STATUS_SOCIAL:** en esta clase se almacena información de las personas según el estatus social de estas, la estructura de esta clase es la siguiente:

```
(defclass STATUS_SOCIAL (is-a PERSONA)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance status_social1 of STATUS_SOCIAL (nombre "pobre"))
(make-instance status_social2 of STATUS_SOCIAL (nombre "rico"))
(make-instance status_social3 of STATUS_SOCIAL (nombre "millonario"))
(make-instance status_social4 of STATUS_SOCIAL (nombre "méndigo"))
(make-instance status_social5 of STATUS_SOCIAL (nombre "miserable"))
(make-instance status_social6 of STATUS_SOCIAL (nombre "potentado"))
(make-instance status_social7 of STATUS_SOCIAL (nombre "indingente"))
```

- **RAZA:** en esta clase se almacena información de las personas según la raza a la que pertenece, la estructura de esta clase es la siguiente:

```
(defclass RAZA (is-a PERSONA)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

(make-instance raza3 of RAZA (nombre "mulato"))
(make-instance raza4 of RAZA (nombre "ladino"))
(make-instance raza5 of RAZA (nombre "indígena"))
(make-instance raza8 of RAZA (nombre "aborigen"))
(make-instance raza9 of RAZA (nombre "autóctono"))
(make-instance raza10 of RAZA (nombre "nativo"))
(make-instance raza11 of RAZA (nombre "natural"))

- **NACIONALIDAD:** en esta clase se almacena información de las personas según la nacionalidad de las mismas, la estructura de esta clase es la siguiente:

(defclass NACIONALIDAD (is-a PERSONA)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))

Algunas de las instancias de la clase son las siguientes:

(make-instance nacionalidad1 of NACIONALIDAD (nombre "extranjero"))
(make-instance nacionalidad2 of NACIONALIDAD (nombre "ciudadano"))
(make-instance nacionalidad3 of NACIONALIDAD (nombre "peregrino"))
(make-instance nacionalidad4 of NACIONALIDAD (nombre "residente"))
(make-instance nacionalidad5 of NACIONALIDAD (nombre "foráneo"))
(make-instance nacionalidad6 of NACIONALIDAD (nombre "forastero"))

- **PERSONA_EDAD:** en esta clase se almacena información de las personas según la edad, la estructura de esta clase es la siguiente:

(defclass PERSONA_EDAD (is-a PERSONA)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))

Algunas de las instancias de la clase son las siguientes:

```
(make-instance edad1 of PERSONA_EDAD (nombre "niño"))
(make-instance edad2 of PERSONA_EDAD (nombre "adulto"))
(make-instance edad3 of PERSONA_EDAD (nombre "anciano"))
(make-instance edad4 of PERSONA_EDAD (nombre "bebé"))
(make-instance edad5 of PERSONA_EDAD (nombre "adolescente"))
```

- ***ESTADO_CIVIL*:** en esta clase se almacena información de las personas según el estado civil de estas. La estructura de esta clase es la siguiente:

```
(defclass ESTADO_CIVIL (is-a PERSONA)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance estado_civil1 of ESTADO_CIVIL (nombre "casado"))
(make-instance estado_civil2 of ESTADO_CIVIL (nombre "viudo"))
(make-instance estado_civil3 of ESTADO_CIVIL (nombre "divorciado"))
(make-instance estado_civil4 of ESTADO_CIVIL (nombre "soltero"))
(make-instance estado_civil5 of ESTADO_CIVIL (nombre "comprometido"))
```

- **PERSONA_ENFERMA:** en esta clase se almacena información de las personas según el estado de salud en el que se encuentre, la estructura de esta clase es la siguiente:

```
(defclass PERSONA_ENFERMA (is-a PERSONA)
        (role concrete)
        (pattern-match reactive)
        (slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance persona_enferma1 of PERSONA_ENFERMA (nombre "minusválidos"))
(make-instance persona_enferma5 of PERSONA_ENFERMA (nombre "discapacitado"))
(make-instance persona_enferma6 of PERSONA_ENFERMA (nombre "estado de
  interdicción"))
(make-instance persona_enferma7 of PERSONA_ENFERMA (nombre "discapacitado"))
```

Y con esto se ha terminado todos los posibles objetos que se activarán cuando se mencione o hable de persona. Una clase superior a persona es ser humano, cuando se hable de ser humano, activara a todas las instancias de persona dado que una persona es un ser humano, a continuación se vera la estructura de la misma:

- **Ser humano:** en esta clase se define la estructura que deben tener las instancias de ser humano, sabiendo que una persona es un *ser humano, pero un ser humano no es una persona*.

```
(defclass SER_HUMANO (is-a USER)
(role  concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

El slot *"nombre"* guarda el nombre de la instancia ser_humano. Algunas de las instancias de la clase son las siguientes:

```
(make-instance ser1 of SER_HUMANO (nombre "ser humano"))
(make-instance ser2 of SER_HUMANO (nombre "feto"))
(make-instance ser3 of SER_HUMANO (nombre "ser concebido"))
(make-instance ser4 of SER_HUMANO (nombre "embrión"))
```

- **Garantías del estado:** esta clase permite almacenar las garantías que proporciona el estado a la persona. Almacenarlas de esta forma, da la

opción a que crezca la base de conocimientos en el futuro. La estructura de la clase es la siguiente:

```
(defclass GARANTIA_ESTADO(is-a USER)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance VIDA of GARANTIA_ESTADO (nombre "vida"))
(make-instance LIBERTAD of GARANTIA_ESTADO (nombre "libertad"))
(make-instance JUSTICIA of GARANTIA_ESTADO (nombre "justicia"))
(make-instance SEGURIDAD of GARANTIA_ESTADO (nombre "seguridad"))
(make-instance PAZ of GARANTIA_ESTADO (nombre "paz"))
(make-instance DESARROLLO_INTEGRAL of GARANTIA_ESTADO (nombre "desarrollo
 integral"))
(make-instance INTEGRIDAD of GARANTIA_ESTADO (nombre "integridad"))
```

- **Familia:** esta clase permite almacenar los sinónimos de la palabra familia. La estructura de la clase es la siguiente:

```
(defclass FAMILIA(is-a USER)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance familia1 of FAMILIA (nombre "familia"))
(make-instance familia2 of FAMILIA (nombre "hogar"))
(make-instance familia3 of FAMILIA (nombre "matrimonio"))
(make-instance familia4 of FAMILIA (nombre "nucleo familiar"))
```

- **Autoridad Judicial:** esta clase permite almacenar las autoridades judiciales competentes; almacenarlas de esta forma, da la opción a que crezca la base de conocimientos en el futuro. La estructura de la clase es la siguiente:

```
(defclass AUTORIDAD_JUDICIAL (is-a AUTORIDAD)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance autoridad_judicial1 of AUTORIDAD_JUDICIAL (nombre "jueces de paz"))
(make-instance autoridad_judicial2 of AUTORIDAD_JUDICIAL (nombre "jueces de narcoactividad"))
(make-instance autoridad_judicial3 of AUTORIDAD_JUDICIAL (nombre "jueces de delitos contra el ambiente"))
(make-instance autoridad_judicial4 of AUTORIDAD_JUDICIAL (nombre "jueces de primera instancia"))
(make-instance autoridad_judicial5 of AUTORIDAD_JUDICIAL (nombre "jueces de ejecución"))
(make-instance autoridad_judicial6 of AUTORIDAD_JUDICIAL (nombre "tribunales de sentencia"));
```

- **Delito:** esta clase permite almacenar los nombres de los que en Guatemala se consideran delitos. La estructura de la clase es la siguiente:

```
(defclass DELITO(is-a USER)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance delito1 of DELITO (nombre "delito"))
(make-instance delito2 of DELITO (nombre "homicidio simple"))
(make-instance delito3 of DELITO (nombre "homicidios calificados"))
(make-instance delito4 of DELITO (nombre "aborto"))
(make-instance delito5 of DELITO (nombre "agresión"))
(make-instance delito6 of DELITO (nombre "lesiones"))
```

- **Falta:** esta clase permite almacenar los nombres de los que en Guatemala se consideran faltas así come el tipo de falta. La estructura de la clase es la siguiente:

```
(defclass FALTA (is-a INFRACCION)
(role concrete)
(pattern-match reactive)
(slot nombre (create-accessor write))
(slot tipo (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance falta1 of FALTA (nombre "faltas contra las personas"))
(make-instance falta2 of FALTA (nombre "faltas contra la propiedad"))
(make-instance falta3 of FALTA (nombre "faltas contra las buenas costumbres"))
(make-instance falta4 of FALTA (nombre "faltas contra los intereses generales"))
```

- **Negación:** esta clase permite almacenar los sinónimos de negaciones, esto se utiliza en las reglas por ej. *No, ningún, etc.,*. La estructura de la clase es la siguiente:

```
(defclass NEGACION (is-a USER)
        (role concrete)
        (pattern-match reactive)
        (slot nombre (create-accessor write)))
```

Algunas de las instancias de la clase son las siguientes:

```
(make-instance NO of NEGACION (nombre no))
(make-instance NINGUN of NEGACION (nombre ningun))
(make-instance NADA of NEGACION (nombre nada))
(make-instance NADIE of NEGACION (nombre nadie))
```

- **FALTO_LIBERTAD:** esta clase almacena los nombres de los que han sido privados de ciertas cosas, como lo son los detenidos o presos.

4.4 Reglas definidas para el proceso de equiparación

Dado que es necesario que exista un proceso de comparación o correspondencia entre las instancias de oración y lo que se desea obtener de ellas, se definen las reglas que ayudan a este proceso.

Una regla esta compuesta de dos partes un si y un entonces, en la parte del si se especifica lo que debe cumplir la regla y en la parte del entonces se le especifica las acciones que debe tomar si se cumple la regla, nótese que únicamente se ejecutara la parte del entonces si se cumple la parte del si por ej.

"Si todos los hombres son libres entonces es constitucional"

El Sistema Experto Legislativo divide las reglas en 2 partes:

- Reiteraciones: son las posibles repeticiones de leyes ya existentes; estas bien son constitucionales, lo único que hacen es ampliar la comprensión de la ley.
- Antinomias: es la contradicción entre la iniciativa de ley y la Constitución de la República.
- Al no ser ninguna de las dos anteriores posiblemente sea un artículo que no abarque los artículos 1-10 tratados en este prototipo.

Los artículos que se abarcan en este prototipo es del 1 -10 de los cuales del 1 al 5 tratan de la persona, sus derechos, el estado y las obligaciones que

tiene este hacia la persona; los siguientes 5 artículos tratan del detenido, sus derechos y garantías.

Antes de definir las reglas se explicará las funciones que sirven para realizar la búsqueda dentro de las instancias de las clases.

Básicamente se usan dos funciones, que ambas retornan TRUE si se encontró elemento o dato buscado y FALSE de lo contrario.

Otra funcionalidad que tienen las funciones es resolver el problema del plural, dado que en la base de hechos se definieron las instancias en singular, esto permite reducir la base de hechos a la mitad debido a que no hay que ingresarlas a la base.

A continuación se muestra el código de las funciones:

```
(deffunction existe_dato (?clase $?a)
(bind ?existe FALSE)
(bind ?i 1)
(bind ?tamanio (length ?a))
(while (<= ?i ?tamanio)
(bind ?valor_lista (nth$ ?i ?a))
(do-for-all-instances ((?p1 ?clase))
(neq (str-index ?p1:nombre ?valor_lista) FALSE)
(bind ?existe TRUE))
(bind ?i (+ ?i 1)))
      ?existe)
```

Se envían dos parámetros para realizar la búsqueda, el primero es la clase en donde se realizará la búsqueda y el segundo la lista que se desea

comparar con la clase, esta lista se examina elemento por elemento con las instancias de la clase.

```
(deffunction existe (?valor $?dato)
(bind ?existe FALSE)
(bind ?i 1)
        (bind ?tamanio (length ?dato))
        (while (<= ?i ?tamanio)
                (bind ?valor_lista (nth$ ?i ?dato))
                (if (eq (str-index ?valor ?valor_lista) 1)
                then
                        (bind ?existe TRUE))
                (bind ?i (+ ?i 1)))
        ?existe)
```

Esta función realiza una búsqueda de un valor en una lista, además que verifica el plural.

Estas 2 funciones ayudarán a comprender las reglas, debido a que las búsquedas se realizan únicamente por medio de estas.

Se puede iniciar con las explicaciones de las reglas después de esta breve explicación.

4.4.1 Reiteraciones de la ley

Se define a continuación todas las reglas encontradas de los artículos 1 al 10 de la Constitución de la República, se muestra el axioma que las define y luego la definición de la regla en pseudocódigo.

- **Artículo 1** del artículo 1 salieron 2 reglas

  La primera regla ha sido definida de la siguiente manera:

"Si el estado de Guatemala se organiza para proteger a la persona o a la familia entonces agregar a lista de constitucionales el artículo 1"

En código la regla se ha definido de la siguiente forma:

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)
(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto
nil)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (existe "organizar" ?v))
(test (existe "proteger" ?p))
(test (or (existe_dato PERSONA ?p) (existe_dato FAMILIA ?p)))
=>
(bind ?existe_articulo (member$ 1 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 1))))
```

En donde ?s es cualquier sinónimo de estado de Guatemala
y ?p es cualquier instancia de persona o cualquier sinónimo de FAMILIA

La segunda regla ha sido definida de la siguiente manera:
"Si la finalidad del estado es el bien común entonces agregar a lista de constitucionales el artículo 1"

En código la regla se ha definido de la siguiente forma:

```
(defrule articulo1b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo
$?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo
nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
```

```
(test (existe "finalidad" ?s))
(test (existe "ser" ?v))
(test (existe_dato BIEN_COMUN ?p))
=>
(bind ?existe_articulo (member$ 1 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 1))))
```

En donde ?s es cualquier sinónimo de estado de Guatemala
y ?p es cualquier instancia de bien común.

- **Artículo 2** la regla definida para el artículo 2 es la siguiente:
"Si estado garantiza (la vida, el desarrollo, educación…) a la persona entonces agregar a lista de constitucionales el artículo 2."

En código la regla se ha definido de la siguiente forma:

```
(defrule articulo2
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)
(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto
nil)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (existe "garantizar" ?v))
(test (existe_dato PERSONA ?p))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 2))))
```

En donde ?s, es cualquier instancia de ESTADO.

Dado a que ?p es un multislot, para que cumpla con la regla debe venir de la siguiente forma:
?p, cualquier o todas las garantías que proporciona el estado
?p, es cualquier instancia de persona.

La segunda regla ha sido definida de la siguiente forma:
"Toda persona tiene derecho a garantias. Siendo estas garantias las que el estado ofrece."

En código la regla se ha definido de la siguiente forma:

```
(defrule articulo2a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe "derecho" ?p))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 2))))
```

- **Artículo 3** la regla definida para el artículo 3 es la siguiente:
  "?s garantiza y/o protege vida humana ?p"

"Si el estado garantiza y/o protege la vida humana del ser humano entonces agregar a lista de constitucionales el artículo 3."

En código la regla se ha definido de la siguiente forma:

```
(defrule articulo3
(object (is-a ORACION) (sustantivo_sujeto $?s) (verbo_aux nil) (verbo $?v) (condicional nil) (sustantivo_predicado $?p) (neg_verbo nil) (neg_sustantivo_sujeto nil) (neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (or (existe "garantizar" ?v) (existe "proteger" ?v)))
(test (existe "vida humana" ?p))
(test (existe_dato SER_HUMANO ?p))
=>
(bind ?existe_articulo (member$ 3 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 3))))
```

En donde:

?s algún sinónimo de Estado

?p es alguna instancia de ser humano

- **Artículo 4** las reglas definidas para este artículo son 4 las cuales se explican a continuación:

  La primera regla ha sido definida de la siguiente manera:

  "Si en Guatemala todos los seres humanos son libres y/o iguales en dignidad y/o derechos entonces agregar a lista de constitucionales el artículo 4."

  En código la regla se ha definido de la siguiente forma:

```
(defrule articulo4a
 (object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo
 $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo
 nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
 (test (existe "guatemala" ?s))
 (test (existe_dato SER_HUMANO ?s))
 (test (existe "ser" ?v))
 (test (or (existe "libres" ?p) (existe "iguales" ?p)))
 (test (or (existe "dignidad" ?p) (existe_dato DERECHO ?p)))
 =>
 (bind ?existe_articulo (member$ 4 ?*lista_c*))
 (if (eq ?existe_articulo FALSE)
 then
 (bind ?*lista_c* (insert$ ?*lista_c* 1 4))))
```

en donde ?s es cualquier instancia de ser humano

?p es cualquier instancia de DERECHO.

La segunda regla ha sido definida de la siguiente manera:

"Si la persona tiene iguales oportunidades y/o responsabilidades entonces agregar a lista de constitucionales el artículo 4."

En código la regla se ha definido de la siguiente forma:

```
(defrule articulo4b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo
$?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo
nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe "iguales" ?p))
```

```
(test (or (existe "oportunidades" ?p) (existe "responsabilidades" ?p)))
=>
(bind ?existe_articulo (member$ 4 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 4))))
```

en donde ?s es cualquier instancia de persona

La tercera regla ha sido definida de la siguiente manera:
"Si ninguna persona es sometida a servidumbre entonces agregar a lista de constitucionales el artículo 4."

En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo4c
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?sn)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?sn) (existe_dato NEGACION ?nv)))
(test (existe_dato PERSONA ?s))
(test (existe "ser" ?v))
(test (existe "sometida" ?p))
(test (existe "servidumbre" ?p))
=>
(bind ?existe_articulo (member$ 4 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 4))))
```

En donde ?n es cualquier instancia de negación por ejemplo no, ninguna, ningún, etc., y ?s es cualquier instancia de persona. Un ejemplo de esta regla puede ser *Ninguna mujer puede ser sometida a servidumbre.*

La cuarta regla ha sido definida de la siguiente manera:
"Si los seres humanos guardan conducta fraternal entre si entonces *agregar a lista de constitucionales el artículo 4."*

En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo4d
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato SER_HUMANO ?s))
(test (existe "deben" ?v))
(test (existe "guardar" ?p))
(test (existe "conducta fraternal" ?p))
=>
(bind ?existe_articulo (member$ 4 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 4))))
```

Se ha terminado la parte de la persona, ahora se comenzara a describir las reglas que tratan acerca del detenido.

- **Artículo 5** la regla definida para el artículo 5 es la siguiente:
  "Si toda persona tiene derecho a hacer lo que la ley no prohibe entonces agregar a lista de constitucionales el artículo 5."

En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo5
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado ?pn))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe_dato "derecho" ?p))
(test (existe_dato "ley" ?p))
(test (existe_dato "prohibe" ?p))
(test (existe_dato NEGACION ?pn))
=>
(bind ?existe_articulo (member$ 5 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 5))))
```

En dónde ?s cualquier instancia de la clase persona,
?v debe tener el verbo en infinitivo “tener”
?pn es cualquier instancia de la clase negación, y
?p siendo un multislot debe tener en los slots los siguientes valores: derecho, ley y prohíbe.
Un ejemplo de esta regla es la iniciativa de ley: “Todo hombre tiene derecho a hacer lo que la ley no prohíbe”

- **Artículo 6:** la reglas definidas para el artículo 6 son las siguientes:
  La primera regla ha sido definida de la siguiente manera:
  “Si ninguna persona puede ser detenida o presa sino por causa de delito y/o falta entonces agregar a lista de constitucionales el artículo 6.”

En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo6a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional ?c)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto ?sn)(neg_sustantivo_predicado nil))
(test (existe_dato NEGACION ?sn))
(test (existe_dato PERSONA ?s))
(test (existe "ser" ?v))
(test (existe_dato FALTO_LIBERTAD ?p))
(test (existe "sino" ?c))
(test (or (existe_dato DELITO ?p) (existe_dato FALTA ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 6))))
```

En donde ?sn es alguna instancia de la clase NEGACION,
?s es alguna instancia de la clase PERSONA,
Siendo ?p un multislot deben contener alguna instancia de las siguientes clases: FALTO_LIBERTAD, DELITO o FALTA.

La segunda regla ha sido definida de la siguiente manera:
*"*Si *el detenido deberá ser puesto únicamente a disposición de la autoridad judicial competente* entonces *agregar a lista de constitucionales el artículo 6."*

En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo6b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "ser" ?v))
(test (existe_dato AUTORIDAD_JUDICIAL ?p))
=>
(bind ?existe_articulo (member$ 6 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 6))))
```

- **Artículo 7** la regla definida para el artículo 7 es la siguiente:
  *"*Si *toda persona detenida debe ser notificada en forma verbal y por* escrito de la causa que motivó su detención, autoridad que la ordenó y lugar en el que permanecerá entonces *agregar a lista de constitucionales el artículo 7."*

  En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo7
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (existe "detenida" ?s))
(test (existe "ser" ?v))
(test (existe "notificada" ?p))
(test (existe "forma" ?p))
```

```
(test (existe "verbal" ?p))
(test (existe "escrito" ?p))
(test (existe "notificada" ?p))
(test (existe "causa" ?p))
(test (existe "autoridad" ?p))
(test (existe "lugar" ?p))
=>
(bind ?existe_articulo (member$ 7 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 7))))
```

- **Artículo 8** las reglas definida para el artículo 8 son las siguientes:

  La primera regla ha sido definida de la siguiente manera:

  *"*Si *todo detenido deberá ser informado de sus derechos* entonces *agregar a lista de constitucionales el artículo 8."*

  En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo8a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenida" ?s))
(test (existe "ser" ?v))
(test (existe "informado" ?p))
(test (existe_dato DERECHO_DETENIDO ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 8))))
```

En donde DERECHO_DETENIDO puede ser cualquier instancia de los derechos que tiene el detenido.

La segunda regla ha sido definida de la siguiente manera:
*"*Si *El detenido NO podrá ser obligado a declarar sino ante autoridad judicial competente* entonces *agregar a lista de constitucionales el artículo 8."*

En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo8b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional ?c)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenida" ?s))
(test (existe_dato NEGACION ?nv))
(test (existe "ser" ?v))
(test (existe "obligado" ?p))
(test (existe "declarar" ?p))
(test (existe "sino" ?c))
(test (existe "ante" ?p))
(test (existe_dato AUTORIDAD_JUDICIAL ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 8))))
```

En donde AUTORIDAD_JUDICIAL puede ser cualquier instancia de autoridad judicial competente.

- **Artículo 9** las reglas definida para el artículo 9 son las siguientes:
  La primera regla ha sido definida de la siguiente manera:
  *"*Si *Las autoridades judiciales son la únicas competentes para interrogar a los detenidos o presos* entonces *agregar a lista de constitucionales el artículo 9."*

  En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo9a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato AUTORIDAD_JUDICIAL ?s))
(test (existe "ser" ?v))
(test (existe "únicas" ?p))
(test (existe " competentes" ?p))
(test (existe "interrogar" ?p))
(test (existe_dato FALTO_LIBERTAD ?p))
=>
(bind ?existe_articulo (member$ 9 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 9))))
```

  En donde ?s es AUTORIDAD_JUDICIAL y esta puede ser cualquier instancia de autoridad judicial competente,
  ?p es FALTO_LIBERTAD, siendo esta detenido o preso.

  La segunda regla ha sido definida de la siguiente manera:
  *"*Si el interrogatorio extrajudicial carece de valor probatorio entonces *agregar a lista de constitucionales el artículo 9."*

En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo9b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "interrogatorio" ?s))
(test (existe "extrajudicial" ?s))
(test (existe "carecer" ?v))
(test (existe "valor" ?p))
(test (existe "probatorio" ?p))
=>
(bind ?existe_articulo (member$ 9 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 9))))
```

- **Artículo 10** la regla definida para el artículo 10 es la siguiente:

  *"*Si *las personas aprehendidas por la autoridad no podrán ser conducidas a lugares de detención, arresto o prisión diferentes a los que están legal y públicamente destinados al efecto* entonces *agregar a lista de constitucionales el artículo 7.*

  En código la regla ha sido definido de la siguiente forma:

```
(defrule articulo10
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "persona aprehendida" ?s))
(test (existe_dato NEGACION ?nv))
(test (existe "poder" ?v))
```

```
(test (existe "conducidas" ?p))
(test (existe_dato LUGAR_DETENCION ?p))
(test (existe "diferentes" ?p))
(test (existe "destinados" ?p))
(test (existe "efecto" ?p))
=>
(bind ?existe_articulo (member$ 10 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 10))))
```

En donde LUGAR_DETENCION es alguna instancia de los lugar de detención, y NEGACION es alguna instancia de la clase NEGACION.

Se ha terminado de explicar las reglas de esta sección, si alguna de las reglas anteriormente mencionadas se activa, se guarda en la variable global de nombre ?*lista_c* el número del artículo de la regla que se activo.

4.4.2 Antinomias

La finalidad de estas reglas es que una iniciativa de ley equipare con alguna de estas, esto para detectar la contradicción de la iniciativa de ley con la Constitución, estas reglas fueron definidas contradiciendo lo que la Constitución dice es decir si la constitución dice: “Todos las personas son libres en derechos” la regla se hizo “No todas las personas son libres en derechos”; sabiendo hay una infinidad de permutaciones para jugar con las palabras, en este prototipo se definen las más comunes.

- **Artículo 1**

  Este artículo trata de que el Estado de Guatemala se organiza para proteger a la persona y a la familia y su finalidad es el bien común.

  Lo que se trata de contradecir es básicamente 2 premisas:

  - el estado de Guatemala se organiza para proteger a la persona y a la familia, y
  - la finalidad del estado es el bien común.

  Las premisas contradichas quedan así:

  - el estado de Guatemala *no* se organiza para proteger a la persona y/o la familia
  - la finalidad del estado no es el bien común

  Sabiendo que cuando se habla de persona es cualquier instancia de la clase PERSONA; del estado de Guatemala es cualquier sinónimo de estado, bien común es cualquier instancia de BIEN_COMUN y *no* es cualquier sinónimo de negación.

  En código las reglas han quedado así:

```
(defrule garantia_estado_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns))
(test (existe "organizar" ?v))
(test (existe "proteger" ?v))
(test (or (existe_dato PERSONA ?p) (existe_dato FAMILIA ?p)))
=>
(bind ?existe_articulo (member$ 1 ?*lista_i*))
```

```
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 1))))


(defrule finalidad_estado_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (existe "finalidad" ?s))
(test (existe "ser" ?v))
(test (existe_dato BIEN_COMUN ?p))
=>
(bind ?existe_articulo (member$ 1 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_c* (insert$ ?*lista_c* 1 1))))
```

- **Artículo 2**

  Este artículo trata de los deberes del estado hacia la persona.

  Lo que se contradice es la siguiente premisa:

  - es deber del Estado garantizarle a los habitantes de la República la vida, la libertad, la justicia,..

  Las premisas contradichas quedan así:

  - no es deber del Estado garantizarle a los habitantes de la República la vida, la libertad, la justicia
  - el estado no garantiza a los habitantes de la República la vida, la libertad, la justicia.
  - Las personas no tienen garantías.

Sabiendo que cuando se habla de libertad, justicia, vida, etc son las garantías que ofrece el estado a la persona y todas estas están incluidas en las instancias de la clase GARANTIA_ESTADO.

En código la regla ha quedado definida de la siguiente manera:

```
(defrule garantias_estado_negadas
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?nv)))
(test (existe_dato ESTADO ?s))
(test (existe "garantizar" ?v))
(test (existe_dato PERSONA ?p))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 2))))

(defrule personas_sin_garantias
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
```

```
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 2))))
```

- **Artículo 3**

En este artículo se enuncia la garantía y la protección del derecho a la vida del ser humano.

Lo que se contradice es la siguiente premisa:

- es deber del Estado garantizarle a los habitantes de la República la vida, la libertad, la justicia,..

Las premisas contradichas quedan así

- el estado no garantiza la vida humana desde su concepción
- el estado no protege la vida humana desde su concepción

Sabiendo que un ser concebido es un ser humano, la regla se aplica para el SER HUMANO.

En código la regla ha sido definida de la siguiente manera:

```
(defrule estado_garantiza_vida_negación
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (existe_dato NEGACION ?nv))
(test (or (existe "garantizar" ?v) (existe "proteger" ?v)))
(test (existe "vida" ?p))
(test (existe "humana" ?p))
(test (existe_dato SER_HUMANO ?p))
=>
(bind ?existe_articulo (member$ 3 ?*lista_i*))
```

```
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 3))))
```

- **Artículo 4**

  Este artículo afirma la libertad e igualdad entre los seres humanos no importando su estado civil.

  Lo que se contradice son las siguientes premisas:

  - en Guatemala todos los seres humanos son libres e iguales en dignidad y derechos.
  - el hombre y la mujer, cualquiera que sea su estado civil, tienen iguales oportunidades y responsabilidades
  - los seres humanos deben guardar conducta fraternal entre si

  Las premisas contradichas quedan así:

  - no todos los seres humanos son libres e iguales en dignidad y derechos
  - los seres humanos no son libres e iguales en dignidad y derechos
  - la(s) persona(s), cualquiera que sea su estado civil, tienen iguales oportunidades y responsabilidades.
  - los seres humanos no deben guardar conducta fraternal entre si

  El estado civil se activa al momento de hacer mención de persona por ejemplo si se habla de un casado es una persona por lo tanto se activara esta regla.

  En código las reglas han sido definidas de la siguiente manera:

```
(defrule libertad_igualdad_negacion
```

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato SER_HUMANO ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe "ser" ?v))
(test (or (existe "libres" ?p) (existe "iguales" ?p)))
(test (or (existe "dignidad" ?p) (existe "derechos" ?p)))
=>
(bind ?existe_articulo (member$ 4 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 4))))
(defrule estado_civil_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe "tener" ?v))
(test (existe "igual" ?p))
(test (or (existe "oportunidad" ?p) (existe "responsabilidad" ?p)))
=>
(bind ?existe_articulo (member$ 4 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 4))))

(defrule conducta_fraternal_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(adverbio ?a)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
```

```
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe_dato SER_HUMANO ?s))
(test (existe "deben" ?v))
(test (existe "guardar" ?a))
(test (existe "conducta" ?p))
(test (existe "fraternal" ?p))
=>
(bind ?existe_articulo (member$ 4 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 4))))
```

- **Artículo 5**

  Este artículo afirma la libertad de acción.

  Lo que se contradice son las siguientes premisas:

  - Toda persona tiene derecho a hacer lo que la ley no prohíbe.

  La premisa contradicha queda así

  - Toda persona tiene derecho a hacer lo que la ley prohíbe.

  En código la regla ha sido definida de la siguiente manera:

```
(defrule ley_prohibe_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado ?pn))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe_dato "derecho" ?p))
(test (existe_dato "ley" ?p))
(test (existe_dato "prohíbe" ?p))
```

```
=>
(bind ?existe_articulo (member$ 5 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 5))))
```

- **Artículo 6**

  Este artículo trata de la detención legal

  Lo que las reglas contradicen son las siguientes premisas:

  - ninguna persona puede ser detenida o presa, sino por causa de delito o falta.
  - los detenidos deberán ser puestos a disposición a disposición de la autoridad judicial competente.

  La premisa contradicha queda así

  - ninguna persona puede ser detenida o presa, sino por causa de algo que no sea delito o falta.
  - los detenidos no deben ser puestos a disposición de la autoridad judicial competente.
  - los detenidos deben ser puestos a disposición de otra que no sea autoridad judicial competente.

  En código la regla ha sido definida de la siguiente manera:

```
(defrule detencion_legal_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional ?c)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto ?sn)(neg_sustantivo_predicado nil))
(test (existe_dato NEGACION ?sn))
(test (existe_dato PERSONA ?s))
(test (existe "ser" ?v))
(test (existe_dato FALTO_LIBERTAD ?p))
```

```
(test (existe "sino" ?c))
(test (or (not (existe_dato DELITO ?p)) (not (existe_dato FALTA ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 6))))

(defrule detencio_legal_negacion1
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "ser" ?v))
(test (not (existe_dato AUTORIDAD_JUDICIAL ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 6))))

(defrule detencion_legal_negacion2
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe_dato NEGACION ?nv))
(test (existe "ser" ?v))
(test (not (existe_dato AUTORIDAD_JUDICIAL ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
```

```
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 6))))
```

- **Artículo 8**

  Este artículo trata de los derechos del detenido.

  Lo que las reglas contradicen son las siguientes premisas:

  - todo detenido deberá ser informado inmediatamente de sus derechos.
  - el detenido no podrá ser obligado a declarar sino ante autoridad judicial competente.

  Las premisas contradichas quedan así

  - todo detenido puede ser informado de sus derechos
  - el detenido no podrá declarar ante autoridad judicial competente

  En código las reglas definidas anteriormente se visualizan así:

```
(defrule detenido_derechos_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "poder" ?v))
(test (existe "ser" ?v))
(test (existe "informado" ?p))
(test (existe_dato DERECHO_DETENIDO ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 8))))
```

```
(defrule declara_autoridad_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "poder" ?v))
(test (existe "declarar" ?v))
(test (existe "ante" ?p))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe_dato AUTORIDAD_JUDICIAL ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 8))))
```

- **Artículo 9**

  Este artículo trata del interrogatorio a detenido o preso.

  Lo que las reglas contradicen son las siguientes premisas:

  - las autoridades judiciales son las únicas competentes para interrogar a los detenidos o preso
  - el interrogatorio extrajudicial carece de valor probatorio

  Las premisas contradichas quedan así

  - las autoridades judiciales no son la únicas competentes para interrogar a los detenidos o presos, ó
  - no son las autoridades judiciales las únicas competentes para interrogar a los detenidos o presos.
  - el interrogatorio extrajudicial no carece de valor probatorio

En código las reglas han sido definidas de la siguiente manera:

```
(defrule interrogatorio_detenidos_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(adverbio ?a)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato AUTORIDAD_JUDICIAL ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe "ser" ?v))
(test (existe "unicas competentes" ?p))
(test (existe "interrogar" ?p))
(test (existe_dato FALTO_LIBERTAD ?p))
=>
(bind ?existe_articulo (member$ 9 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 9))))

(defrule interrogatorio_extrajudicial_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "interrogatorio" ?s))
(test (existe "extrajudicial" ?s))
(test (existe_dato NEGACION ?nv))
(test (existe "carecer" ?v))
(test (existe "valor" ?p))
(test (existe "probatorio" ?p))
=>
(bind ?existe_articulo (member$ 9 ?*lista_i*))
```

```
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 9))))
```

- **Artículo 10**

  Este artículo trata del Centro de detención legal.

  Lo que la regla contradice es la siguiente premisa:

  - las personas aprehendidas por la autoridad no podrán ser conducidas a lugares de detención, arresto o prisión diferentes a los que están legal y públicamente destinados al efecto.

  La premisa contradicha queda así:

  - las personas aprehendidas por la autoridad podrán ser conducidas a lugares de detención, arresto o prisión diferentes a los que están legal y públicamente destinados al efecto.

  En código la regla ha sido definida de la siguiente manera:

```
(defrule lugar_detencion_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "persona aprehendida" ?s))
(test (existe "poder" ?v))
(test (existe "conducidas" ?p))
(test (existe_dato LUGAR_DETENCION ?p))
(test (existe "diferentes" ?p))
(test (existe "destinados" ?p))
(test (existe "efecto" ?p))
=>
```

```
(bind ?existe_articulo (member$ 10 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then
(bind ?*lista_i* (insert$ ?*lista_i* 1 10))))
```

Si alguna de las reglas descritas en esta sección se activan, se guarda en la variable global ?*lista_c* el número del artículo de la regla que se activo.

4.5 Generación del resultado

Luego de realizarse la equiparación y de que los hechos que conforman la iniciativa de ley activarán determinadas reglas, se esta en disposición de saber si la propuesta de ley es constitucional o inconstitucional, recordando que los resultados que proporciona el Sistema Experto Legislativo son una ayuda al legislador, para que el mismo pueda tomar sus propias decisiones.

Si no se activo ninguna regla que detectara antinomias o reiteraciones a la Constitución, la salida que proporciona el Sistema Experto Legislativo en pantalla y específicamente al usuario es: “La iniciativa de ley no se encuentra entre el dominio de los articulos 1 – 10 de la Constitución de la república”, de lo contrario el sistema indicará claramente la razón por la cual ha considerado que la iniciativa de ley es inconstitucional, es decir, el sistema indica los artículos de la constitución encontrados para los que la iniciativa de ley contradice en su dominio o que la iniciativa de ley es constitucional aunque tenga reiteraciones a las leyes ya existentes, mostrando los artículos que se activaron.

La regla que se utiliza para proporcionar los resultados es la siguiente:

“Si se han terminado de leer las oraciones entonces comparar la lista de inconstitucionales si existe más de algún elemento en esta lista indicar los

artículos que se encontraron en donde la propuesta de ley los contradecía; de lo contrario comparar la lista de constitucionales, si existe más de algún elemento en esta lista indicar el número de artículo que se encontró en donde la iniciativa de ley lo abarcaba; de lo contrario si en ninguna de las 2 listas se encontró valores indicar que la iniciativa de ley esta fuera de los rangos de ley definidos."

En código la regla ha sido definido de la siguiente forma:

```
(defrule validar
(declare (salience -1000))
(fin fin)
=>
(bind ?a (length ?*lista_i*))
(bind ?b (length ?*lista_c*))
(if (> ?a 0)
then
(printout t "Advertencias y/o sugerencias" crlf)
(printout t "El proyecto de ley se encontro inconsitucional en base a los siguientes articulos:" crlf)
(printout t "Articulo(s): ")
(bind ?aux_a 1)
(while (<= ?aux_a ?a)
(bind ?val_art (nth$ ?aux_a ?*lista_i*))
(printout t ?val_art", ")
(bind ?aux_a (+ ?aux_a 1)))
else
(if (> ?b 0)  then
(printout t "Advertencias y/o sugerencias" crlf)
(printout t "El proyecto de ley se encontro consitucional en base a los siguientes articulos:" crlf)
```

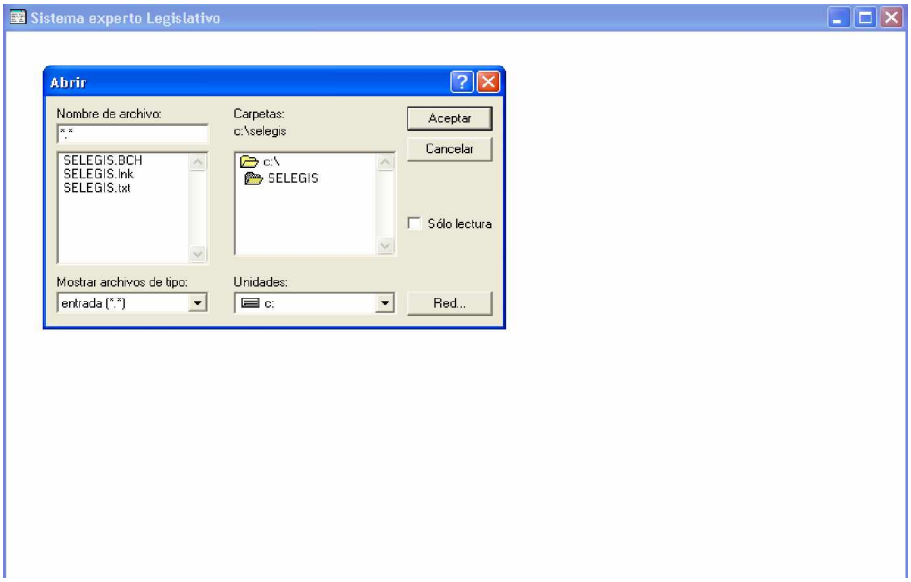
```
(printout t "Articulo(s): ")
(bind ?aux_b 1)
(while (<= ?aux_b ?b)
(bind ?val_art (nth$ ?aux_b ?*lista_c*))
(printout t ?val_art", ")
(bind ?aux_b (+ ?aux_b 1)))
else
(printout t "La información ingresada no se encuentra dentro de los articulos 1-10 de la" crlf)
(printout t "Constitución de la República de Guatemala." crlf))
(printout t crlf)))
```

4.6 Evaluación de un caso

A continuación se describe el procedimiento que debe seguir el usuario para ingresar al sistema la iniciativa de ley.

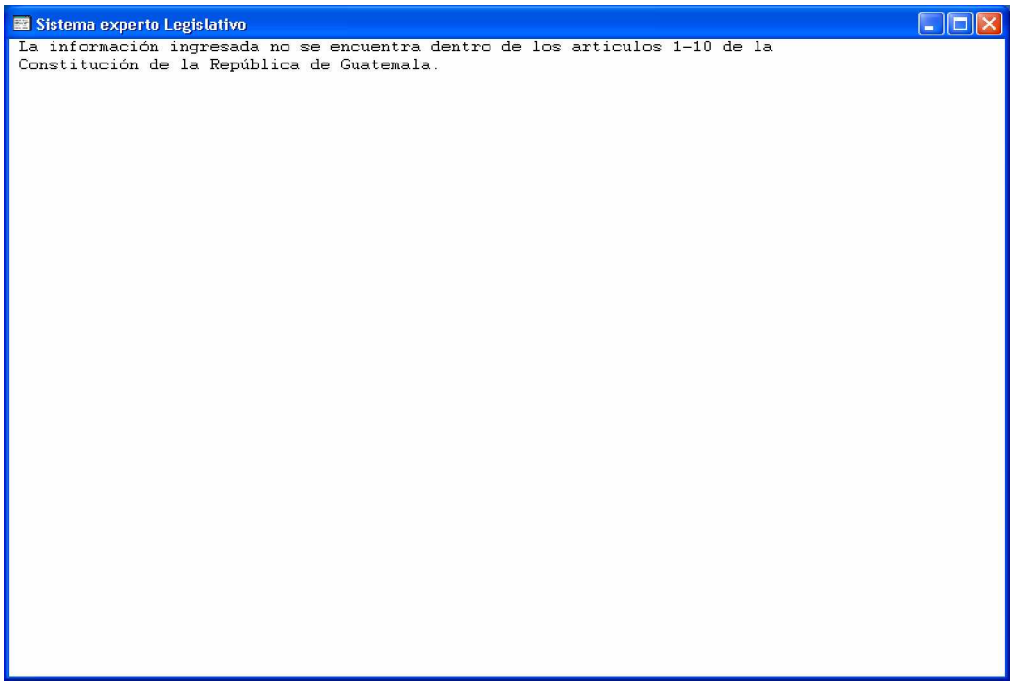
Grabar en la raíz C la carpeta SELEGIS, dentro de esa carpeta esta el ejecutable llamado con el mismo nombre, con el ratón pulsarlo dos veces luego le aparecerá una pantalla como se muestra en la figura 4,

Figura 4. Archivo de Entrada



Escoger el archivo de entrada, debe recordar que este archivo debe de tener la estructura de la oracion. Luego el sistema equiparara la iniciativa de ley con los artículos expuestos y le mostrara en pantalla el resultado como se muestra en la figura 5:

Figura 5. Resultados Obtenidos



A continuación se detalla el proceso que sigue el sistema experto legislativo cuando le es ingresada una nueva instancia de *oracion* que en este caso es la iniciativa de ley.

4.6.1 CASO No.1

A continuación son descritos los pasos.

1. Luego de que es ingresada una iniciativa de ley que en este caso es: "*El estado se organiza para proteger a los niños*" y haber definido a la misma en una instancia, como por ejemplo:

   ([oracion1] of ORACION
   (sustantivo_sujeto "estado")
   (verbo_aux nil)
   (verbo_modal nil)
   (verbo "organizar")

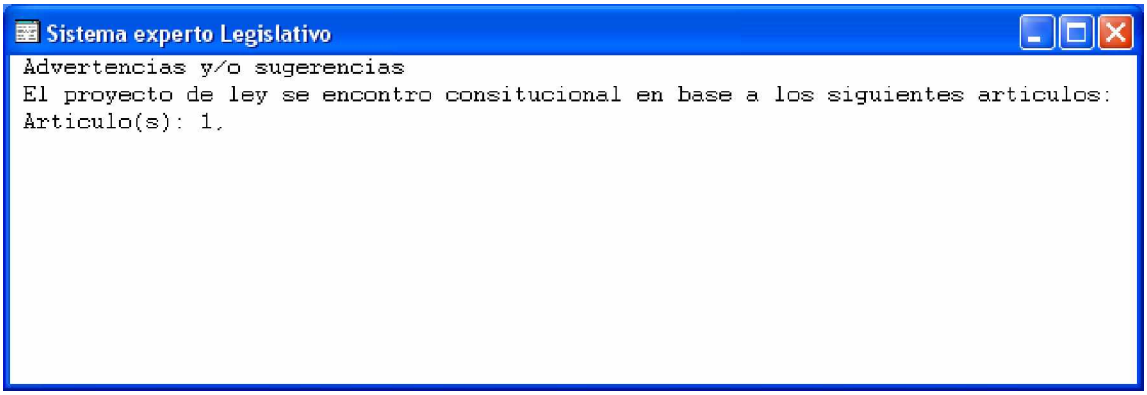
(adverbio nil)
(condicional nil)
(sustantivo_predicado "proteger" "niños")
(neg_verbo nil)
(neg_sustantivo_sujeto nil)
(neg_sustantivo_predicado nil))

2. Comienza el proceso de equiparación entre todas las reglas, y la regla que activa es:
   La regla con nombre *articulo1a,* esta regla dicen que si una instancia de oración entra por ellas es porque la propuesta de ley es constitucional y es ingresado el artículo a la lista de constitucionalidades.

3. Se verifica que no existan más instancias de oración.

4. Al momento de no existir más instancias de oración se ejecuta la regla *validar*, la funcionalidad de esta regla es revisar los datos que existen en las listas tanto de constitucionalidades como inconstitucionalidades, dándole prioridad a la lista de inconstitucionalidades porque si hay mas de algún valor en esta lista es porque no es constitucional la iniciativa de ley de lo contrario es revisada la lista de constitucionalidades y si encuentra más de algún valor en ella es porque la iniciativa de ley es constitucional, como este caso es en este punto donde se activa y muestra en pantalla que la iniciativa de ley es constitucional en base al artículo 1 de la Constitución de la República de Guatemala, como se muestra en la figura 6.

Figura 6. Caso No. 1, resultados



4.6.2 CASO No.2

A continuación son descritos los pasos.

1. Luego de que es ingresada una iniciativa de ley que en este caso es: *"Ningún niño tendrá derecho a la vida*" y tener definida esta oración en una instancia por ejemplo:
   ([oracion2] of ORACION
   (sustantivo_sujeto "niño")
   (verbo_aux nil)
   (verbo_modal nil)
   (verbo "tener")
   (adverbio nil)
   (condicional nil)
   (sustantivo_predicado "derecho""vida")
   (neg_verbo nil)
   (neg_sustantivo_sujeto "ningun")
   (neg_sustantivo_predicado nil))

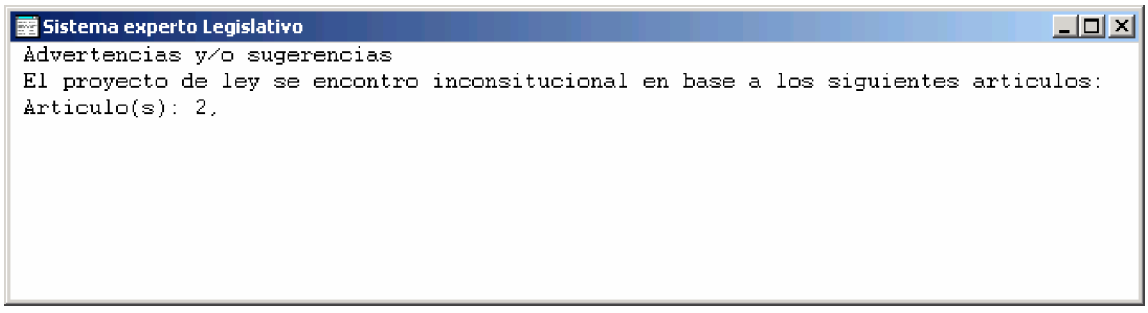
2. Comienza el proceso de equiparación entre todas las reglas, y la regla que activa es:

   La regla con nombre *personas_sin_garantias,* esta regla dicen que si una instancia de oración entra por ellas es porque la propuesta de ley es inconstitucional y es ingresado el artículo a la lista de inconstitucionalidades.

3. Se verifica que no existan más instancias de oración.

4. Al momento de no existir más instancias de oración se ejecuta la regla *validar*, la funcionalidad de esta regla es revisar los datos que existen en las listas tanto de constitucionalidades como inconstitucionalidades, dándole prioridad a la lista de inconstitucionalidades porque si hay mas de algún valor en esta lista es porque no es constitucional la iniciativa de ley y es aquí donde se conoce que la iniciativa de ley ingresada es inconstitucional en base al articulo 2 de la Constitución de la República de Guatemala, a continuación se muestra lo que aparece en pantalla.

Figura 7. Caso No. 2, resultados



4.6.3 CASO No. 3

A continuación son descritos los pasos.

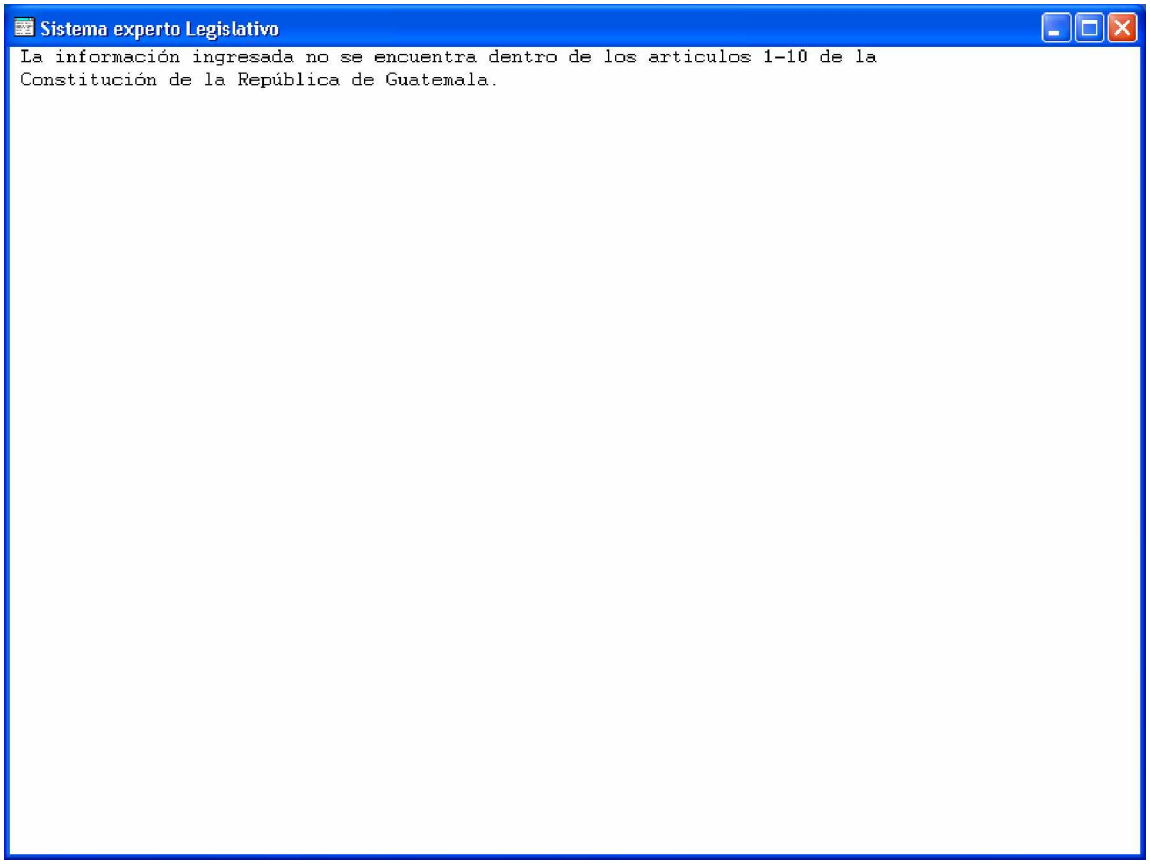
1. Luego de que es ingresada una iniciativa de ley que en este caso es: "*Todos los presos tendrán derechos a la libertad* " y tener definida esta oración en una instancia por ejemplo:
   ([FRASE] of ORACION
   (sustantivo_sujeto "presos")
   (verbo_aux nil)
   (verbo_modal nil)
   (verbo tener)
   (adverbio)
   (condicional nil)
   (sustantivo_predicado "derecho" "libertad")
   (neg_verbo nil)
   (neg_sustantivo_sujeto nil)
   (neg_sustantivo_predicado nil))

2. Comienza el proceso de equiparación entre todas las reglas y dado a que esta oración no pertenece a ninguno de los articulos 1 -10 entonces no se activa ninguna regla.

3. Al momento de no existir más instancias de oración se ejecuta la regla *validar*, la funcionalidad de esta regla es revisar los datos que existen en la listas tanto de constitucionalidades como inconstitucionalidades, dándole prioridad a la lista de inconstitucionalidades porque si hay mas de algún valor en esta lista es porque no es constitucional la iniciativa de ley de lo

contrario es revisada la lista de constitucionalidades y si encuentra más de algún valor en ella es porque la iniciativa de ley es constitucional, y por ultimo si no cumple con ninguna de las anteriores es por que la iniciativa de ley ingresada no se encuentra dentro del rango de los artículos de este sistema experto por lo tanto se activa hasta la regla y tira el resaltado en pantalla como a continuación se muestra.

Figura 8. Caso No. 3, resultados



4.7 Requerimientos para la implementación

Para la implementación del Sistema Experto Legislativo los requerimientos básicos son:

4.7.1 Hardware

Los requerimientos mínimos para el funcionamiento óptimo del sistema son y su desarrollo exitoso consiste como mínimo en una computadora con un procesado Pentium II y con memoria RAM de 128 MB y un disco duro de 20 GB, este último requerimiento dependerá del crecimiento a futuro de la base de conocimientos.

4.7.2 Software

Para el desarrollo del sistema se utilizo la herramienta de programación de Inteligencia Artificial EHSIS, el cual se ejecuta en ambiente Windows, este mismo sistema operativo es requerido para la implementación del Prototipo de Sistema Experto Legislativo.

4.7.3 Recursos humanos

Para la implementación de este sistema es necesario que la persona que realizará el desarrollo del mismo tenga conocimiento básico de lo que es la legislación guatemalteca y un conocimiento extenso en lo que ha Sistemas Experto se refiere, para poder implementar el mismo en un ambiente de desarrollo ideal, como lo es EHSIS.

Por otra parte es esencial que la persona que hará uso de este sistema sea un conocedor de la legislación guatemalteca y a la vez sepa hacer uso de la computadora y programas en ambiente Windows.

5 ANOTACIONES FINALES Y PERSPECTIVAS DEL SISTEMA EXPERTO

Al terminar con la descripción de la funcionalidad del prototipo de Sistema Experto Legislativo, se pretende dar a conocer los avances y sugerencias para que un futuro sea posible continuar con la construcción del mismo.

Con este trabajo se ha pretendido romper un paradigma de los sistemas, ya que, es conocido por la mayoría de personas que cuentan con el conocimiento apropiado, que los sistemas abiertos, como lo son: el derecho y las ciencias sociales son difíciles de modelar y, aún más, encapsularse en un sistema de cómputo. Con lo cual se hallado con este prototipo un punto intermedio, en el cual no es encerrado y delimitado el funcionamiento de todas las variables que interactúan en el derecho, ni tampoco se ha encontrado un punto final en el cual se halla podido modelar con exactitud el comportamiento social de las leyes, sino que al igual que una persona aprende a través de su experiencia, su entorno y la información recibida, el prototipo ha sido construido cimentando las bases para su posterior crecimiento, el cual será añadido por los expertos interesados en la madurez del mismo, como se detalla en las siguientes secciones de este capítulo.

5.1 Impacto del sistema

Este Sistema Experto como tal cuenta con un gran número de posibles aplicaciones, obviamente la mayoría de ellas se encuentra enfocadas en el derecho legislativo, principalmente. Sin embargo este sistema comienza por sentar las bases para el encuentro entre sistemas abiertos, como lo es el

derecho y la Inteligencia Artificial, a fin de generar procesos de automatización y apoyo en la toma de decisiones.

5.1.1 En la enseñanza

Para la enseñanza, este puede utilizarse en alumnos que están estudiando derecho y comienzan la carrera a fin de que su análisis en las leyes, comience a ser desarrollado en sus primeras etapas de aprendizaje en dicha área, principalmente en el la inducción, la cual es importante para la toma de decisiones en su futuro profesional.

A la vez es importante este sistema para alumnos que comienzan sus estudios de Inteligencia Artificial, ya que en este sistema son utilizados conceptos y técnicas de dicha rama de estudio y específicamente de los Sistemas Expertos. Además de la inteligencia artificial es importante para personas que necesiten conocer acerca de la programación orientada a objetos, ya que en este sistema fue utilizada este tipo de programación al hacerse uso de objetos para definir elementos importantes de la Constitución.

5.1.2 En el campo profesional

En la actualidad han tomado auge los sistemas de apoyo a la toma de decisiones, sin embargo estos recurren constantemente al uso de complejas y costosas arquitecturas y metodologías, además de que su uso es principalmente necesario en escenarios bajo un mayor control, como lo son las estadísticas y más concretamente cifras numéricas, sin embargo, a menudo no se recurre a sistemas que apoyen la toma de decisiones que requieren de un mayor análisis y racionalización, como lo es el área de derecho y legislación, en estos campos de estudio se torna importante este trabajo para apoyar la toma

de decisiones de los expertos en las áreas de estudio anteriormente mencionadas, como tal, estos sistemas no pueden sustituir a los expertos, debido a la complejidad de los procesos mentales que deben ser realizados para tomar una decisión por parte de los abogados, legisladores, etc..

5.1.3 En la informática

En la actualidad no ha sido explorada una posible relación entre la Inteligencia Artificial y el Derecho Constitucional en Guatemala, de cierto, existen sistemas de cómputo que automatizan algunas tareas que realizan los abogados, pero aun no se ha roto la brecha entre la IA y el Derecho Constitucional como lo demuestran otras naciones, como lo son Argentina y España. Con el Sistema Experto Legislativo se ha explorado esta brecha existente entre sistemas tan poco estudiados juntos como el derecho y la inteligencia artificial, sentándose así las bases para el desarrollo de un sistema experto legislativo que fundamente su funcionamiento en la Inteligencia Artificial.

5.2 Limites del prototipo

A continuación son expuestos los límites que fueron encontrados durante la construcción del prototipo de sistema experto legislativo:

- El análisis realizado por los abogados es demasiado complejo, tiene demasiadas relaciones que trasladan el análisis de una conclusión hacia otra, para llegar a una conclusión final, sería necesario construir un sistema experto demasiado complejo, con lo cual se incurriría en demasiado tiempo, costos y la ayuda de varios expertos para tener una base de conocimientos lo suficientemente sólida. La lógica utilizada en

este Sistema Experto Legislativo sirve como una base para futuros trabajos y solamente es un prototipo de 10 artículos.

- Si una frase planteada por el legislador es compleja en su significado, el sistema muy difícilmente la entendería, por ejemplo *"Las autoridades judiciales son las únicas competentes para interrogar a los detenidos o presos. Esta diligencia deberá practicarse dentro de un plazo que no exceda de veinticuatro horas"*, con la primera oración no tendría problema el sistema de reconocerla y entenderla, sin embargo con la segunda oración si tendría problema de entenderla, porque habla de *"esta diligencia"* que esta haciendo referencia a la oración anterior lo cual no es interpretado por el sistema, esta es una modificación que debe ser considerada para mejorar el prototipo de este trabajo de graduación.

- Las reglas que fueron estructuradas en el sistema fueron colocadas de tal manera que describieran cada uno de los artículos correspondientes de la constitución, sin embargo, estos artículos es posible consolidarse en ocasiones en una sola regla. Para este sistema no fue hecho de esta manera ya que se perdería el nivel de detalle al que llega el sistema a fin de proporcionarle información al usuario del artículo en que se encontró la constitucionalidad o inconstitucionalidad. Por ejemplo, la autoridad judicial y sus funciones pudieron resumirse en una sola regla, pero fue preferible para este sistema separarlas en los distintos artículos que las abarcan para poder proporcionarle al usuario una descripción especifica del artículo donde se encontró la antinomia o reiteración de la ley.

5.3 Aportes futuros

El prototipo de sistema experto legislativo, enunciado en este trabajo de graduación, no pretende ser un sistema final, aunque si completo, por lo cual a continuación son listados atributos para continuar con el desarrollo del prototipo o para crear nuevos sistemas expertos, que sean similares o que enriquezcan al prototipo de Sistema Experto Legislativo.

- Para tener una base de conocimientos robusta, debe existir una parte del Sistema Experto Legislativo que interrogue a diferentes personas que sean expertas en la materia, para saber el punto de vista de las mismas y obtener diferentes criterios de decisión, por ejemplo: para una abogado *"bien común"* puede significar una cosa mientras que para otro abogado puede tener un significado distinto, de este modo sería un aporte importante que el Sistema almacene ambos criterios y sirvan para tomar un decisión más acertada. Con esto se deduce que la creación de una base de datos y su interrelación con el SE puede ser un buen punto de partida para continuar con el crecimiento y madurez de este prototipo. EHSIS proporciona las herramientas para la conexión de la base de datos, lo restante es agregar marcos que no estén definidos en este trabajo de graduación y relacionarlos, luego crear la interfaz para que el usuario (legislador) ingrese su conocimientos y el significado que para el tiene el mismo. Lo que se lograría con esto es hacer más robusto el trabajo de graduación realizado y en la parte de deducción el sistema se enriquece.
- Este trabajo se enfoca específicamente en la vista de un legislador es decir como el legislador interpreta un proyecto de ley. Otra forma de enriquecer este trabajo es agregar la opción en la que el legislador ingrese una palabra de la que desee saber todas los

artículos que tengan relación con la misma, entonces el sistema debe mostrara en pantalla todas las posibles opciones, algo parecido a un buscador. Las relaciones ya se encuentran en este trabajo de graduación al igual que las reglas lo único que hace falta es la interfaz para el legislador ingrese su petición y guardar en la base los artículos completos. Con esto se puede ir jugando se puede mostrar las palabras sinónimas, palabras que se deducen de otra, las palabras padre de otra por ej. Para el concepto padre, el concepto superior es persona y, así, puede continuar.

- Otra forma de incrementar o poder continuar con el Sistema Experto Legislativo es identificar las relaciones que existen entre los diferentes conceptos, encontrados en la constitución, dado que haciendo esto se podrán definir más reglas que intercepten cuando una iniciativa de ley es constitucional o inconstitucional.

CONCLUSIONES

1. Se creó un prototipo, de Sistema Experto Legislativo completamente, funcional, el cual, basándose en los artículos del 1 al 10 de la Constitución de la República de Guatemala, es capaz de verificar la constitucionalidad o inconstitucionalidad en una iniciativa de ley; con lo cual se ha verificado la validez de este tipo de sistemas en el apoyo a la toma de decisiones del legislador.

2. A pesar de la inexistencia de Sistemas Expertos de esta naturaleza en la República de Guatemala se pudo demostrar que la construcción es factible de realizar con leyes guatemaltecas.

3. Aún, cuando en Guatemala no existen sistemas en los cuales la información referente al derecho constitucional haya sido modelada, a través de un sistema de cómputo, fue posible representar este tipo de conocimiento; a través de una metodología de marcos, en la cual cada marco abstrae una porción del conocimiento obtenido a través de los artículos 1 al 10 de la Constitución de la República de Guatemala.

4. Los conceptos doctrinarios legales están interrelacionados entre sí por medio de diferentes tipos de relaciones, por ejemplo: relaciones de herencia, asignación, etc.

5. El prototipo de Sistema Experto proporciona al legislador un resultado donde especifica si es una reiteración o una antinomia especificando los artículos de donde se baso para dar tal resultado.

RECOMENDACIONES

1. Sistemas de esta índole son pocos los desarrollados y en nuestro país, inexistentes, por tal razón, es necesario que se apoye a la creación e implementación de este tipo de sistemas, tanto en el financiamiento de los mismos como en la asignación de expertos que estén dispuestos a proporcionar tal conocimiento.

2. Las universidades deberían enseñarles a los futuros profesionales a utilizar sistemas de toma de decisión desde su formación académica, ya sea a través de sistemas, previamente, construidos o a través del análisis, diseño y construcción de sistemas de este tipo.

3. Implementar sistemas basados en Marcos dado que estos son los que mas se adecuan a la realidad.

BIBLIOGRAFÍA

1. Alberto Pereira Orozco. Introducción al Estudio del Derecho 1. Universidad de San Carlos de Guatemala, Facultad de Ciencias Jurídicas y Sociales, 2000.

2. Ashley, K.K., "Modelling legal argument: Reasoning with cases and hypotheticals", Cambridge: MIT Press, 1990.

3. Borja, Rodrigo. *Enciclopedia de la Política*. 2ª, Edición, Editorial Progreso S.A., México, D.F. ,1998.

4. Carlos Vasquez Ortiz. Derecho Civil 1. (Guatemala: Editorial Crockmen)

5. Código Penal, Decreto No. 17-73

6. Diccionario Jurídico Espasa (España: Editorial Espasa Calpe, S.A.)

7. Hurtado Aguilar, Hernán. *Derecho Penal Compensado, Comentarios a la Parte General del Código Penal*. Editorial Landívar. Guatemala, 1984. Págs.

8. Ramón García y Gross. Pequeño Larousse Ilustrado. (México: Editorial Offset Larios, 1979)

9. Ramiro de León Carpio. **Catecismo Constitucional**.(7ª. Edición; Guatemala: Tipografía Nacional,1995)

10. Ossorio, Manuel. *Diccionario de Ciencias Jurídicas y Sociales.* Editorial Heliasta, Buenos Aires, Argentina, 1987.

11. Popple. J., "SHYSTER: A pragmatic legal expert system", Unpublised Phd diss., Departament of Computer Science, Australian National University, 1993.

12. Porrúa Pérez, Francisco. *Teoría del Estado.* Editorial Porrúa, 1954.

13. Puig Peña, Federico. *Compendio de Derecho Civil Español*. Tomo I, Madrid, España, Ediciones Pirámide, 1976.

14. Sonia Yolanda Castañeda Ramírez, "Implementación de Sistemas expertos utilizando frames", Universidad de San Carlos de Guatemala, Tesis, 2001

PAGINAS VISITADAS EN INTERNET

15. Diccionario de la Real Academia Española, http://www.rae.es/

16. Inteligencia Artificial Distribuida y Razonamiento Basado en Casos en la Arquitectura de un Sistema Basado en el Conocimiento para la Educación a Distancia (SBC-ED), revista No. 9, http://www.inf.udec.cl/revista/

17. Mercedes García de Quesada, “Estructura Definicional Terminográfica en el subdominio de la oncología clínica”. Universidad de Granada, Tesis, septiembre de 2004. [Documento disponible en la red: http://elies.rediris.es/elies14/index.html]

18. Monografías, http://www.monografias.com/trabajos/iartificial/pagina4_5.htm

19. Monografías, http://www.monografias.com/trabajos11/teosis/teosis.shtml#ABIER

20. http://www.ieid.org/congreso/ponencias/Cuadrado%20Gamarra,%20Nuria.pdf.

21. http://elies.rediris.es/elies14/index.html

22. http://www.alfa-redi.org/revista/data/48-8.asp

23. http://www.itba.edu.ar/capis/webcapis/RGMITBA/comunicacionesrgm/c-UnSistemaExpertoLegal-SID-JAIO2001.pdf

24. http://www.fi.uba.ar/laboratorios/ lsi/jucse01-argumentacion.pdf

APÉNDICE

CÓDIGO FUENTE EN EHSIS DEL SISTEMA EXPERTO LEGISLATIVO:

```
(reset)
;en esta clase se guarda todas las instancias de ser_humano por ejempo: embrion que a la vez una persona
;es un ser humano
(defclass SER_HUMANO (is-a USER)
  (role  concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
;------------------------DE LA PERSONA
(defclass PERSONA (is-a SER_HUMANO)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass MIEMBRO_FAMILIA (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass PROFESIONAL (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)

(defclass STATUS_SOCIAL (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
```

```
)

(defclass RAZA (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass NACIONALIDAD (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)

(defclass PERSONA_EDAD (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)

(defclass ESTADO_CIVIL (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass PERSONA_ENFERMA (is-a PERSONA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)

;---------------------------------------------------------------------------
(defclass ESTADO(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
```

```
)
;se guardan las instancias de diferentes garantias que ofrece el estado
(defclass GARANTIA_ESTADO(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
  )

;se guardan los nombres de cosas que mensocaben la dignidad de una persona
(defclass MENOSCABAR_DIGNIDAD(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot dignidad (create-accessor write))
  )

(defclass FALTO_LIBERTAD(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
  )
;se guardan los nombres de los que en Guatemala se consideran delitos
(defclass DELITO(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass INFRACCION (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
;se guardan los nombres de los que en Guatemala se consideran faltas, asi como el tipo de falta por ej. faltas a los reglamentos
(defclass FALTA (is-a INFRACCION)
  (role concrete)
```

```
  (pattern-match reactive)
  (slot nombre (create-accessor write))
  (slot tipo (create-accessor write))
)
(defclass AUTORIDAD (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
; en los artículos estudiados la autoridad judicial es la mencionada de las autoridades en el art.  que dice quienes
;conforman la autoridad_judicial esto sin llegar a tanto detalle
(defclass AUTORIDAD_JUDICIAL (is-a AUTORIDAD)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass AUTORIDAD_POLICIAL (is-a AUTORIDAD)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)

;todos los derechos que son inherentes a la persona, algunos no se cumplen con los detenidos porque no tienen el derecho de
;locomoción p.e., por eso se separaron los derechos del detenido
(defclass DERECHO (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass DERECHO_DETENIDO (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
```

```
)
(defclass DILIGENCIA(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass DILIGENCIA_PENAL(is-a DILIGENCIA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass DILIGENCIA_POLICIAL(is-a DILIGENCIA)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
;se gurdan los lugares permitidos para que permanezca una persona detenida
(defclass LUGAR_DETENCION(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
;los nombres de los diferentes documentos de identificación
(defclass DOCUMENTACION(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
;personas que pueden participar en la diligencia penal
(defclass PERSONA_DILIGENCIA(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass NEGACION (is-a USER)
```

```
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass FAMILIA(is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass BIEN_COMUN (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot nombre (create-accessor write))
)
(defclass ORACION (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (multislot sustantivo_sujeto (create-accessor write))
  (slot verbo_aux (create-accessor write))
  (slot verbo_modal (create-accessor write))
  (multislot verbo (create-accessor write))
  (multislot adverbio (create-accessor write))
  (slot condicional (create-accessor write))
  (multislot sustantivo_predicado (create-accessor write))
  (slot neg_verbo (create-accessor write))
  (slot neg_sustantivo_sujeto (create-accessor write))
  (slot neg_sustantivo_predicado (create-accessor write))
)
(deffacts finalizar
  (fin nil)
)
(set-strategy breadth)
(make-instance NO of NEGACION (nombre "no"))
(make-instance NINGUN of NEGACION (nombre "ningun"))
(make-instance NADA of NEGACION (nombre "nada"))
```

```
(make-instance NADIE of NEGACION (nombre "nadie"))

(make-instance autoridad_judicial1 of AUTORIDAD_JUDICIAL (nombre "jueces de paz"))
(make-instance autoridad_judicial2 of AUTORIDAD_JUDICIAL (nombre "jueces de narcoactividad"))
(make-instance autoridad_judicial3 of AUTORIDAD_JUDICIAL (nombre "jueces de delitos contra el ambiente"))
(make-instance autoridad_judicial4 of AUTORIDAD_JUDICIAL (nombre "jueces de primera instancia"))
(make-instance autoridad_judicial5 of AUTORIDAD_JUDICIAL (nombre "jueces de ejecución"))
(make-instance autoridad_judicial6 of AUTORIDAD_JUDICIAL (nombre "tribunales de sentencia"))
(make-instance autoridad_judicial7 of AUTORIDAD_JUDICIAL (nombre "salas de la corte de apelaciones "))
(make-instance autoridad_judicial8 of AUTORIDAD_JUDICIAL (nombre "Juez de delitos contra el ambiente"))
(make-instance autoridad_judicial9 of AUTORIDAD_JUDICIAL (nombre "corte de suprema justicia"))
(make-instance autoridad_judicial10 of AUTORIDAD_JUDICIAL (nombre "juez de ejecución"))
(make-instance autoridad_judicial11 of AUTORIDAD_JUDICIAL (nombre "autoridad judicial"))
(make-instance autoridad_judicial12 of AUTORIDAD_JUDICIAL (nombre "autoridad judicial competente"))
(make-instance autoridad_judicial13 of AUTORIDAD_JUDICIAL (nombre "Juez de paz"))
(make-instance autoridad_judicial14 of AUTORIDAD_JUDICIAL (nombre "Juez de narcoactividad"))
(make-instance autoridad_judicial15 of AUTORIDAD_JUDICIAL (nombre "tribunal de sentencia"))
(make-instance autoridad_judicial16 of AUTORIDAD_JUDICIAL (nombre "sala de la corte de apelación"))

(make-instance derecho1 of DERECHO (nombre "derecho"))
(make-instance derecho2 of DERECHO (nombre "vida"))
```

```
(make-instance derecho3 of DERECHO (nombre "nombre"))
(make-instance derecho4 of DERECHO (nombre "familia"))
(make-instance derecho5 of DERECHO (nombre "propiedad privada"))
(make-instance derecho6 of DERECHO (nombre "estudio"))
(make-instance derecho7 of DERECHO (nombre "trabajo"))
(make-instance derecho8 of DERECHO (nombre "libre locomoción"))
(make-instance derecho9 of DERECHO (nombre "libertad de pensamiento"))
(make-instance derecho10 of DERECHO (nombre "acudir a los tribunales"))
(make-instance derecho11 of DERECHO (nombre "libertad de culto"))
(make-instance derecho12 of DERECHO (nombre "matrimonio"))



(make-instance VIDA of GARANTIA_ESTADO (nombre "vida"))
(make-instance LIBERTAD of GARANTIA_ESTADO (nombre "libertad"))
(make-instance JUSTICIA of GARANTIA_ESTADO (nombre "justicia"))
(make-instance SEGURIDAD of GARANTIA_ESTADO (nombre "seguridad"))
(make-instance PAZ of GARANTIA_ESTADO (nombre "paz"))
(make-instance DESARROLLO_INTEGRAL of GARANTIA_ESTADO (nombre "desarrollo
integral"))
(make-instance INTEGRIDAD of GARANTIA_ESTADO (nombre "integridad"))

(make-instance DETENIDO of FALTO_LIBERTAD (nombre detenido))
(make-instance PRESO of FALTO_LIBERTAD (nombre preso))

(make-instance estado1 of ESTADO (nombre "estado"))
(make-instance estado2 of ESTADO (nombre "estado de guatemala"))

(make-instance bien_comun1 of BIEN_COMUN (nombre "bien comun"))

(make-instance ser1 of SER_HUMANO (nombre "ser humano"))
(make-instance ser2 of SER_HUMANO (nombre "feto"))
(make-instance ser3 of SER_HUMANO (nombre "ser concebido"))
(make-instance ser4 of SER_HUMANO (nombre "embrión"))
```

```
(make-instance persona1 of PERSONA (nombre "hombre"))
(make-instance persona2 of PERSONA (nombre "personas de la tercera edad"))
(make-instance persona3 of PERSONA (nombre "persona"))
(make-instance persona4 of PERSONA (nombre "habitante"))
(make-instance persona5 of PERSONA (nombre "mujer"))
(make-instance persona6 of PERSONA (nombre "varon"))
(make-instance persona7 of PERSONA (nombre "hembra"))
(make-instance persona8 of PERSONA (nombre "niño"))


(make-instance miembro1 of MIEMBRO_FAMILIA (nombre "tio"))
(make-instance miembro2 of MIEMBRO_FAMILIA (nombre "abuelo"))
(make-instance miembro3 of MIEMBRO_FAMILIA (nombre "hermano"))
(make-instance miembro4 of MIEMBRO_FAMILIA (nombre "primo"))
(make-instance miembro5 of MIEMBRO_FAMILIA (nombre "sobrino"))
(make-instance miembro6 of MIEMBRO_FAMILIA (nombre "nieto"))
(make-instance miembro7 of MIEMBRO_FAMILIA (nombre "bisnieto"))
(make-instance miembro8 of MIEMBRO_FAMILIA (nombre "bisabulo"))
(make-instance miembro9 of MIEMBRO_FAMILIA (nombre "madre"))
(make-instance miembro10 of MIEMBRO_FAMILIA (nombre "padre"))
(make-instance miembro11 of MIEMBRO_FAMILIA (nombre "hijo"))
(make-instance miembro12 of MIEMBRO_FAMILIA (nombre "adoptado"))
(make-instance miembro13 of MIEMBRO_FAMILIA (nombre "conviviente"))
(make-instance miembro14 of MIEMBRO_FAMILIA (nombre "cónyuge"))

(make-instance profesional1 of PROFESIONAL (nombre "profesional"))
(make-instance profesional2 of PROFESIONAL (nombre "doctor"))
(make-instance profesional3 of PROFESIONAL (nombre "licenciado"))
(make-instance profesional4 of PROFESIONAL (nombre "abogado"))
(make-instance profesional5 of PROFESIONAL (nombre "maestro"))
(make-instance profesional6 of PROFESIONAL (nombre "agricultor"))
(make-instance profesional7 of PROFESIONAL (nombre "ama de casa"))
(make-instance profesional8 of PROFESIONAL (nombre "tejedor"))
(make-instance profesional9 of PROFESIONAL (nombre "secretaria"))
(make-instance profesional10 of PROFESIONAL (nombre "programador"))
```

```
(make-instance profesional11 of PROFESIONAL (nombre "presidente"))
(make-instance profesional12 of PROFESIONAL (nombre "tesorero"))
(make-instance profesional13 of PROFESIONAL (nombre "dentista"))
(make-instance profesional14 of PROFESIONAL (nombre "odontologo"))
(make-instance profesional15 of PROFESIONAL (nombre "operador"))
(make-instance profesional16 of PROFESIONAL (nombre "cocinero"))
(make-instance profesional17 of PROFESIONAL (nombre "congresista"))
(make-instance profesional18 of PROFESIONAL (nombre "notario"))
(make-instance profesional19 of PROFESIONAL (nombre "arquitecto"))
(make-instance profesional20 of PROFESIONAL (nombre "comerciante"))
(make-instance profesional21 of PROFESIONAL (nombre "administrador"))
(make-instance profesional22 of PROFESIONAL (nombre "negociante"))
(make-instance profesional23 of PROFESIONAL (nombre "enfermero"))
(make-instance profesional24 of PROFESIONAL (nombre "medico"))
(make-instance profesional25 of PROFESIONAL (nombre "perito"))
(make-instance profesional26 of PROFESIONAL (nombre "minero"))
(make-instance profesional27 of PROFESIONAL (nombre "sacerdote"))
(make-instance profesional28 of PROFESIONAL (nombre "pastor"))
(make-instance profesional29 of PROFESIONAL (nombre "profesión"))
(make-instance profesional30 of PROFESIONAL (nombre "científico"))
(make-instance profesional31 of PROFESIONAL (nombre "matemático"))
(make-instance profesional32 of PROFESIONAL (nombre "físico"))
(make-instance profesional33 of PROFESIONAL (nombre "neurologo"))
(make-instance profesional34 of PROFESIONAL (nombre "naturista"))
(make-instance profesional35 of PROFESIONAL (nombre "ingeniero"))
(make-instance profesional36 of PROFESIONAL (nombre "psicologo"))
(make-instance profesional37 of PROFESIONAL (nombre "carpintero"))
(make-instance profesional38 of PROFESIONAL (nombre "funcionarios"))
(make-instance profesional39 of PROFESIONAL (nombre "empleados públicos"))
(make-instance profesional40 of PROFESIONAL (nombre "juez"))
(make-instance profesional41 of PROFESIONAL (nombre "maquillista"))
(make-instance profesional42 of PROFESIONAL (nombre "nutriologa"))
(make-instance profesional43 of PROFESIONAL (nombre "barbero"))
(make-instance profesional44 of PROFESIONAL (nombre "estilista"))
(make-instance profesional45 of PROFESIONAL (nombre "jardinero"))
```

```
(make-instance profesional46 of PROFESIONAL (nombre "piloto"))
(make-instance profesional47 of PROFESIONAL (nombre "barrendero"))
(make-instance profesional48 of PROFESIONAL (nombre "portero"))
(make-instance profesional49 of PROFESIONAL (nombre "alcalde"))
(make-instance profesional50 of PROFESIONAL (nombre "concejal"))
(make-instance profesional51 of PROFESIONAL (nombre "zapatero"))
(make-instance profesional52 of PROFESIONAL (nombre "lustrador"))
(make-instance profesional53 of PROFESIONAL (nombre "asesor"))
(make-instance profesional54 of PROFESIONAL (nombre "bachiller"))

(make-instance status_social1 of STATUS_SOCIAL (nombre "pobre"))
(make-instance status_social2 of STATUS_SOCIAL (nombre "rico"))
(make-instance status_social3 of STATUS_SOCIAL (nombre "millonario"))
(make-instance status_social4 of STATUS_SOCIAL (nombre "méndigo"))
(make-instance status_social5 of STATUS_SOCIAL (nombre "miserable"))
(make-instance status_social6 of STATUS_SOCIAL (nombre "potentado"))
(make-instance status_social7 of STATUS_SOCIAL (nombre "indingente"))
(make-instance status_social8 of STATUS_SOCIAL (nombre "pordiocero"))
(make-instance status_social9 of STATUS_SOCIAL (nombre "burgués"))
(make-instance status_socia10 of STATUS_SOCIAL (nombre "miserable"))
(make-instance status_socia11 of STATUS_SOCIAL (nombre "mísero"))


(make-instance raza1 of RAZA (nombre "negro"))
(make-instance raza2 of RAZA (nombre "blanco"))
(make-instance raza3 of RAZA (nombre "mulato"))
(make-instance raza4 of RAZA (nombre "ladino"))
(make-instance raza5 of RAZA (nombre "indígena"))
(make-instance raza8 of RAZA (nombre "aborigen"))
(make-instance raza9 of RAZA (nombre "autóctono"))
(make-instance raza10 of RAZA (nombre "nativo"))
(make-instance raza11 of RAZA (nombre "natural"))
(make-instance raza12 of RAZA (nombre "vernáculo"))
(make-instance raza13 of RAZA (nombre "ario"))
(make-instance raza14 of RAZA (nombre "indoeuropeo"))
```

```
(make-instance raza15 of RAZA (nombre "indogermánico"))
(make-instance raza16 of RAZA (nombre "endrino"))
(make-instance raza17 of RAZA (nombre "maya"))
(make-instance raza18 of RAZA (nombre "garfona"))

(make-instance nacionalidad1 of NACIONALIDAD (nombre "extranjero"))
(make-instance nacionalidad2 of NACIONALIDAD (nombre "ciudadano"))
(make-instance nacionalidad3 of NACIONALIDAD (nombre "peregrino"))
(make-instance nacionalidad4 of NACIONALIDAD (nombre "residente"))
(make-instance nacionalidad5 of NACIONALIDAD (nombre "foráneo"))
(make-instance nacionalidad6 of NACIONALIDAD (nombre "forastero"))
(make-instance nacionalidad7 of NACIONALIDAD (nombre "oriundo"))
(make-instance nacionalidad8 of NACIONALIDAD (nombre "domiciliado"))
(make-instance nacionalidad9 of NACIONALIDAD (nombre "persona civil"))
(make-instance nacionalidad10 of NACIONALIDAD (nombre "vecino"))
(make-instance nacionalidad11 of NACIONALIDAD (nombre "doble nacionalidad"))
(make-instance nacionalidad12 of NACIONALIDAD (nombre "multiple nacionalidad"))

(make-instance edad1 of PERSONA_EDAD (nombre "niño"))
(make-instance edad2 of PERSONA_EDAD (nombre "adulto"))
(make-instance edad3 of PERSONA_EDAD (nombre "anciano"))
(make-instance edad4 of PERSONA_EDAD (nombre "bebé"))
(make-instance edad5 of PERSONA_EDAD (nombre "adolescente"))
(make-instance edad6 of PERSONA_EDAD (nombre "longevo"))
(make-instance edad7 of PERSONA_EDAD (nombre "viejo"))
(make-instance edad8 of PERSONA_EDAD (nombre "mayor de edad"))
(make-instance edad9 of PERSONA_EDAD (nombre "menor de edad"))

(make-instance estado_civil1 of ESTADO_CIVIL (nombre "casado"))
(make-instance estado_civil2 of ESTADO_CIVIL (nombre "viudo"))
(make-instance estado_civil3 of ESTADO_CIVIL (nombre "divorciado"))
(make-instance estado_civil4 of ESTADO_CIVIL (nombre "soltero"))
```

(make-instance persona_enferma1 of PERSONA_ENFERMA (nombre "minusválidos"))

(make-instance persona_enferma2 of PERSONA_ENFERMA (nombre "persona que adolecen de limitaciones físicas"))

(make-instance persona_enferma3 of PERSONA_ENFERMA (nombre "persona que adolecen de limitaciones psíquicas"))

(make-instance persona_enferma4 of PERSONA_ENFERMA (nombre "persona que adolecen de limitaciones sensoriales"))

(make-instance persona_enferma5 of PERSONA_ENFERMA (nombre "discapacitado"))

(make-instance persona_enferma6 of PERSONA_ENFERMA (nombre "estado de interdicción"))

(make-instance familia1 of FAMILIA (nombre "familia"))

(make-instance familia2 of FAMILIA (nombre "hogar"))

(make-instance familia3 of FAMILIA (nombre "matrimonio"))

(make-instance familia4 of FAMILIA (nombre "nucleo familiar"))

(make-instance DETENIDO of PERSONA_DILIGENCIA (nombre "detenido"))

(make-instance OFENDIDO of PERSONA_DILIGENCIA (nombre "ofendido"))

(make-instance MINISTERIO_PUBLICO of PERSONA_DILIGENCIA (nombre "ministerio_publico"))

(make-instance ABOGADO of PERSONA_DILIGENCIA (nombre "abogado"))

(make-instance derecho_detenido1 of DERECHO_DETENIDO (nombre "no declar"))

(make-instance derecho_detenido2 of DERECHO_DETENIDO (nombre "tener abogado defensor"))

(make-instance derecho_detenido3 of DERECHO_DETENIDO (nombre "juicio justo"))

(make-instance derecho_detenido4 of DERECHO_DETENIDO (nombre "defensor"))

(make-instance derecho_detenido5 of DERECHO_DETENIDO (nombre "tener proceso legal"))

(make-instance derecho_detenido5 of DERECHO_DETENIDO (nombre "citado oido y vencido en proceso legal"))

(make-instance delito1 of DELITO (nombre "delito"))

(make-instance delito2 of DELITO (nombre "homicidio simple"))

(make-instance delito3 of DELITO (nombre "homicidios calificados"))

```
(make-instance delito4 of DELITO (nombre "aborto"))
(make-instance delito5 of DELITO (nombre "agresión"))
(make-instance delito6 of DELITO (nombre "lesiones"))
(make-instance delito7 of DELITO (nombre "delito deportivo"))
(make-instance delito8 of DELITO (nombre "exposición de personas a peligo"))
(make-instance delito9 of DELITO (nombre "delito contra la seguridad de transito"))
(make-instance delito10 of DELITO (nombre "delitos contra el honor"))
(make-instance delito11 of DELITO (nombre "calumnia"))
(make-instance delito12 of DELITO (nombre "injuria"))
(make-instance delito13 of DELITO (nombre "difamación"))
(make-instance delito14 of DELITO (nombre "delitos contra la libertad"))
(make-instance delito15 of DELITO (nombre "delitos contra el pudor"))
(make-instance delito16 of DELITO (nombre "violación"))
(make-instance delito17 of DELITO (nombre "estrupro"))
(make-instance delito18 of DELITO (nombre "abusos deshonestos"))
(make-instance delito19 of DELITO (nombre "rapto"))
(make-instance delito20 of DELITO (nombre "corrupción de menores"))
(make-instance delito21 of DELITO (nombre "delitos contra la seguridad"))
(make-instance delito22 of DELITO (nombre "delitos contra la libertad individual"))
(make-instance delito23 of DELITO (nombre "allanamiento de morada"))
(make-instance delito24 of DELITO (nombre "sustracción de menores"))
(make-instance delito25 of DELITO (nombre "coacciones"))
(make-instance delito26 of DELITO (nombre "amenazas"))
(make-instance delito27 of DELITO (nombre "violación"))
(make-instance delito28 of DELITO (nombre "revelación de secretos"))
(make-instance delito29 of DELITO (nombre "delitos contra la libertad de cultos"))
(make-instance delito30 of DELITO (nombre "delito de inseminación"))
(make-instance delito31 of DELITO (nombre "delito contra el orden juridico familiar"))
(make-instance delito32 of DELITO (nombre "delito contra el estado civil"))
(make-instance delito33 of DELITO (nombre "celebración de matrimonios ilegales"))
(make-instance delito34 of DELITO (nombre "adulterio"))
(make-instance delito35 of DELITO (nombre "concubinato"))
(make-instance delito36 of DELITO (nombre "incesto"))
(make-instance delito37 of DELITO (nombre "incumplimiento de deberes"))
(make-instance delito38 of DELITO (nombre "delito contra el patrimonio"))
```

```
(make-instance delito39 of DELITO (nombre "hurto"))
(make-instance delito40 of DELITO (nombre "robo"))
(make-instance delito41 of DELITO (nombre "usurpaciónes"))
(make-instance delito42 of DELITO (nombre "extorsión"))
(make-instance delito43 of DELITO (nombre "chantaje"))
(make-instance delito44 of DELITO (nombre "estafa"))
(make-instance delito45 of DELITO (nombre "apropiaciones indebidas"))
(make-instance delito46 of DELITO (nombre "defraudación tributaria"))
(make-instance delito47 of DELITO (nombre "delito contra el derecho de autor"))
(make-instance delito48 of DELITO (nombre "delito contra la propiedad industrial"))
(make-instance delito49 of DELITO (nombre "delitos informaticos"))
(make-instance delito50 of DELITO (nombre "usura"))
(make-instance delito51 of DELITO (nombre "daños"))
(make-instance delito52 of DELITO (nombre "disparo de arma de fuego"))

(make-instance delito52 of DELITO (nombre "incendio"))
(make-instance delito53 of DELITO (nombre "estragos"))
(make-instance delito54 of DELITO (nombre "delito contra los medios de comunicación"))
(make-instance delito55 of DELITO (nombre "delito contra el transporte"))
(make-instance delito56 of DELITO (nombre "piratería"))
(make-instance delito57 of DELITO (nombre "delito contra la salud"))
(make-instance delito58 of DELITO (nombre "delitos contra la fe pública"))
(make-instance delito59 of DELITO (nombre "delitos contra el patrimonio nacional"))
(make-instance delito60 of DELITO (nombre "falsificación de la moneda"))
(make-instance delito61 of DELITO (nombre "falsificación de los documentos"))
(make-instance delito62 of DELITO (nombre "falsificación de sellos"))
(make-instance delito63 of DELITO (nombre "falsificación de papel sellado"))
(make-instance delito64 of DELITO (nombre "falsificación de sellos de correo"))
(make-instance delito65 of DELITO (nombre "falsificación de timbres"))
(make-instance delito66 of DELITO (nombre "depredación del patrimonio nacional"))
(make-instance delito67 of DELITO (nombre "delitos de falsedad personal"))

(make-instance delito68 of DELITO (nombre "delitos contra la economía nacional"))
(make-instance delito69 of DELITO (nombre "delitos contra el comercio"))
(make-instance delito70 of DELITO (nombre "delitos contra la industria"))
```

```
(make-instance delito71 of DELITO (nombre "delitos contra el regimen tributario"))

(make-instance delito72 of DELITO (nombre "traición"))
(make-instance delito73 of DELITO (nombre "espionaje"))

(make-instance delito74 of DELITO (nombre "delitos eleccionarios"))
(make-instance delito75 of DELITO (nombre "delitos de cohecho"))
(make-instance delito76 of DELITO (nombre "negociaciones ilicitas"))
(make-instance delito77 of DELITO (nombre "prevaricación"))

(make-instance delito78 of DELITO (nombre "encubrimiento"))
(make-instance delito79 of DELITO (nombre "juegos ilicitos"))

(make-instance falta1 of FALTA (nombre "faltas contra las personas"))
(make-instance falta2 of FALTA (nombre "faltas contra la propiedad"))
(make-instance falta3 of FALTA (nombre "faltas contra las buenas costumbres"))
(make-instance falta4 of FALTA (nombre "faltas contra los intereses generales"))
(make-instance falta5 of FALTA (nombre "faltas contra el régimen de las poblaciones"))
(make-instance falta6 of FALTA (nombre "faltas contra el orden público"))
(make-instance falta7 of FALTA (nombre "faltas contra el orden jurídico tributario"))
(make-instance falta8 of FALTA (nombre "falta"))
(make-instance falta9 of FALTA (nombre "falta contra la persona"))
(make-instance falta10 of FALTA (nombre "falta contra la propiedad"))
(make-instance falta11 of FALTA (nombre "falta contra los intereses generales"))



; lista que guarda que los articulos que activo la oracion y que son constitucionales
(defglobal
  ?*lista_c* = (create$)
  ?*lista_i* = (create$)
  ?*ventana* = 0
)

(defrule inicio
```

```
(declare (salience 1000))
=>
  (bind ?*ventana* (CreateTextWindow "Sistema experto Legislativo" 25 50 800 600))
  (bind  ?file (OpenFileBox ?*ventana* 32 32 "entrada (*.*)"  "*.*"))
  (load-instances ?file)
 )

;devuelve FALSE si no existe el valor en la clase especificada y TRUE de lo contrario
(deffunction existe_dato (?clase $?a)
  (bind ?existe FALSE)
  (bind ?i 1)
  (bind ?tamanio (length ?a))
  (while (<= ?i ?tamanio)
            (bind ?valor_lista (nth$ ?i ?a))
            (do-for-all-instances ((?p1 ?clase))
            (neq (str-index ?p1:nombre ?valor_lista) FALSE)
            (bind ?existe TRUE))
            (bind ?i (+ ?i 1))
  )
  ?existe
)


(deffunction existe (?valor $?dato)
  (bind ?existe FALSE)
  (bind ?i 1)
  (bind ?tamanio (length ?dato))
  (while (<= ?i ?tamanio)
            (bind ?valor_lista (nth$ ?i ?dato))
            (if (eq (str-index ?valor ?valor_lista) 1)
            then
                        (bind ?existe TRUE)
            )
            (bind ?i (+ ?i 1))
  )
```

```
  ?existe
)

;regla que se cumple al finalizarse de leer la oración
(defrule validar
(declare (salience -1000))
;(fin fin)
=>
 (bind ?a (length ?*lista_i*))
 (bind ?b (length ?*lista_c*))
 (if (> ?a 0)
  then
        (printout ?*ventana* "Advertencias y/o sugerencias" crlf)
        (printout ?*ventana* "El proyecto de ley se encontro inconsitucional en base a los siguientes articulos:" crlf)
        (printout ?*ventana* "Articulo(s): ")
        (bind ?aux_a 1)
        (while (<= ?aux_a ?a)
              (bind ?val_art (nth$ ?aux_a ?*lista_i*))
              (printout ?*ventana* ?val_art", ")
              (bind ?aux_a (+ ?aux_a 1))
        )

  else
        (if (> ?b 0)
          then
              (printout ?*ventana* "Advertencias y/o sugerencias" crlf)
              (printout ?*ventana* "El proyecto de ley se encontro consitucional en base a los siguientes articulos:" crlf)
              (printout ?*ventana* "Articulo(s): ")
              (bind ?aux_b 1)
              (while (<= ?aux_b ?b)
                    (bind ?val_art (nth$ ?aux_b ?*lista_c*))
                    (printout ?*ventana* ?val_art", ")
                    (bind ?aux_b (+ ?aux_b 1))
```

```
                    )
                else
                    (printout ?*ventana* "La información ingresada no se encuentra dentro de los articulos 1-10 de la" crlf)
                    (printout ?*ventana* "Constitución de la República de Guatemala." crlf)
                )
                (printout ?*ventana* crlf)
            )
        )

        ;El estado garantiza y protege la vida humana desde su concepción
        (defrule articulo3
        (object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
        (test (existe_dato ESTADO ?s))
        (test (or (existe "garantizar" ?v) (existe "proteger" ?v)))
        (test (existe "vida humana" ?p))
        (test (existe_dato SER_HUMANO ?p))
        =>
        (bind ?existe_articulo (member$ 3 ?*lista_c*))
        (if (eq ?existe_articulo FALSE)
        then

          (bind ?*lista_c* (insert$ ?*lista_c* 1 3)))
        )


        ;El estado de Guatemala se organiza para proteger a la persona y la familia
        (defrule articulo1a
        (object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
        (test (existe_dato ESTADO ?s))
        (test (existe "organizar" ?v))
```

```
(test (existe "proteger" ?p))
(test (or (existe_dato PERSONA ?p) (existe_dato FAMILIA ?p)))
=>
(bind ?existe_articulo (member$ 1 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 1)))
)



;El estado su finalidad es el bien común ó la finalidad del estado es el bien común
(defrule articulo1b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (existe "finalidad" ?s))
(test (existe "ser" ?v))
(test (existe_dato BIEN_COMUN ?p))
=>
(bind ?existe_articulo (member$ 1 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 1)))
)

;Es deber del Estado garantizarle a los habitantes de la República la vida, la libertad, la justicia ,,
(defrule articulo2
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
```

```
(test (existe "garantizar" ?v))
(test (existe_dato PERSONA ?p))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 2)))
)

(defrule articulo2a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe "derecho" ?p))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 2)))
)



;En Guatemala todos los seres humanos son libres e iguales en dignidad y derechos, duda
(defrule articulo4a
```

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "guatemala" ?s))
(test (existe_dato SER_HUMANO ?s))
(test (existe "ser" ?v))
(test (or (existe "libres" ?p) (existe "iguales" ?p)))
(test (or (existe "dignidad" ?p) (existe_dato DERECHO ?p)))
=>
(bind ?existe_articulo (member$ 4 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 4)))
)

;El hombre y la mujer, cualquiera que sea su estado civil, tienen iguales oportunidades y responsabilidades
(defrule articulo4b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe "iguales" ?p))
(test (or (existe "oportunidades" ?p) (existe "responsabilidades" ?p)))
=>
(bind ?existe_articulo (member$ 4 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
  (bind ?*lista_c* (insert$ ?*lista_c* 1 4)))
)

;Ninguna persona puede ser sometida a servidumbre, falta condicion que mensocabe su dignidad
```

```
(defrule articulo4c
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?sn)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?sn) (existe_dato NEGACION ?nv)))
(test (existe_dato PERSONA ?s))
(test (existe "ser" ?v))
(test (existe "sometida" ?p))
(test (existe "servidumbre" ?p))
=>
(bind ?existe_articulo (member$ 4 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
  (bind ?*lista_c* (insert$ ?*lista_c* 1 4)))
)

;Los seres humanos deben guardar conducta fraternal entre si
(defrule articulo4d
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato SER_HUMANO ?s))
(test (existe "deben" ?v))
(test (existe "guardar" ?p))
(test (existe "conducta fraternal" ?p))
=>
(bind ?existe_articulo (member$ 4 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
  (bind ?*lista_c* (insert$ ?*lista_c* 1 4)))
)
;Toda persona tiene derecho a hacer lo que la ley no prohibe
(defrule articulo5
```

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado ?pn))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe_dato "derecho" ?p))
(test (existe_dato "ley" ?p))
(test (existe_dato "prohíbe" ?p))
(test (existe_dato NEGACION ?pn))
=>
(bind ?existe_articulo (member$ 5 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 5)))
)

;Ninguna persona puede ser detenida o presa, sino por causa de delito o falta y en virtud de orden librada con apego a la ley
;por autoridad judicial competente
(defrule articulo6a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional ?c)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto ?sn)(neg_sustantivo_predicado nil))
(test (existe_dato NEGACION ?sn))
(test (existe_dato PERSONA ?s))
(test (existe "ser" ?v))
(test (existe_dato FALTO_LIBERTAD ?p))
(test (existe "sino" ?c))
(test (or (existe_dato DELITO ?p) (existe_dato FALTA ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then
```

```
  (bind ?*lista_c* (insert$ ?*lista_c* 1 6))
))

;Los detenidos deberan ser puestos a disposición de la autoridad judicial competente en un plazo que no exceda de seis horas
;y no podra quedar sujeto a ninguna otra autoridad
(defrule articulo6b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "ser" ?v))
(test (existe_dato AUTORIDAD_JUDICIAL ?p))
=>
(bind ?existe_articulo (member$ 6 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 6)))

)

;Toda persona detenida deberá ser notificada inmediatamente, en forma verbal y por escrito, de la causa que motivó su detención,
;autoridad que la ordenó y lugar en el que permanecerá.
(defrule articulo7
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (existe "detenida" ?s))
(test (existe "ser" ?v))
(test (existe "notificada" ?p))
(test (existe "forma" ?p))
(test (existe "verbal" ?p))
```

```
(test (existe "escrito" ?p))
(test (existe "notificada" ?p))
(test (existe "causa" ?p))
(test (existe "autoridad" ?p))
(test (existe "lugar" ?p))
=>
(bind ?existe_articulo (member$ 7 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 7)))

)

;Todo detenido deberá ser informado inmediatamente de sus derechos
(defrule articulo8a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "ser" ?v))
(test (existe "informado" ?p))
(test (existe_dato DERECHO_DETENIDO ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 8)))

)

;El detenido no podrá ser obligado a declarar sino ante autoridad judicial competente
(defrule articulo8b
```

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional ?c)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe_dato NEGACION ?nv))
(test (existe "ser" ?v))
(test (existe "obligado" ?p))
(test (existe "declarar" ?p))
(test (existe "sino" ?c))
(test (existe "ante" ?p))
(test (existe_dato AUTORIDAD_JUDICIAL ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 8)))

)

;Las autoridades judiciales son la únicas competentes para interrogar a los detenidos o presos.
(defrule articulo9a
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe_dato AUTORIDAD_JUDICIAL ?s))
(test (existe "ser" ?v))
(test (existe "únicas" ?p))
(test (existe " competentes" ?p))
(test (existe "interrogar" ?p))
(test (existe_dato FALTO_LIBERTAD ?p))
=>
(bind ?existe_articulo (member$ 9 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
```

```
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 9)))

)

;El interrogatorio extrajudicial carece de valor probatorio
(defrule articulo9b
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "interrogatorio" ?s))
(test (existe "extrajudicial" ?s))
(test (existe "carecer" ?v))
(test (existe "valor" ?p))
(test (existe "probatorio" ?p))
=>
(bind ?existe_articulo (member$ 9 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 9)))

)


;Las personas aprehendidas por la autoridad no podrán ser conducidas a lugares de detención, arresto o prisión
;diferentes a los que están legal y pùblicamente destinados al efecto
;sinonimos de aprehendida
(defrule articulo10
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "persona aprehendida" ?s))
```

```
(test (existe_dato NEGACION ?nv))
(test (existe "poder" ?v))
(test (existe "conducidas" ?p))
(test (existe_dato LUGAR_DETENCION ?p))
(test (existe "diferentes" ?p))
(test (existe "destinados" ?p))
(test (existe "efecto" ?p))
=>
(bind ?existe_articulo (member$ 10 ?*lista_c*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_c* (insert$ ?*lista_c* 1 10))
))


;INCOSTITUCIONAL


;El estado de Guatemala no se organiza para proteger a la persona y la familia
(defrule garantia_estado_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (existe "organizar" ?v))
(test (existe "proteger" ?v))
(test (or (existe_dato PERSONA ?p) (existe_dato FAMILIA ?p)))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns))
=>
(bind ?existe_articulo (member$ 1 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 1))
```

```
))



;la finalidad del estado no es el bien comun ...
;El estado su finalidad es el bien comun
(defrule finalidad_estado_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (existe "finalidad" ?s))
(test (existe "ser" ?v))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns))
(test (existe_dato BIEN_COMUN ?p))
=>
(bind ?existe_articulo (member$ 1 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 1)))
)



;No es deber del Estado garantizarle a los habitantes de la Repùblica la vida, la libertad, la justicia ,,
;or el estado no garantiza a los habitantes....
(defrule garantias_estado_negadas
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe_dato ESTADO ?s))
(test (existe "garantizar" ?v))
```

```
(test (existe_dato PERSONA ?p))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 2))
))



;las personas no tienen garantias
(defrule personas_sin_garantias
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
;(test (existe_dato NEGACION ?ns))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe_dato GARANTIA_ESTADO ?p))
=>
(bind ?existe_articulo (member$ 2 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 2))
))



;El estado no garantiza y protege la vida humana desde su concepcion
(defrule estado_garantiza_vida_negación
```

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato ESTADO ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (or (existe "garantizar" ?v) (existe "proteger" ?v)))
(test (existe "vida" ?p))
(test (existe "humana" ?p))
(test (existe_dato SER_HUMANO ?p))
=>
(bind ?existe_articulo (member$ 3 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 3))
))
```

```
;no todos los seres humanos son libres e iguales en dignidad y derechos, o los seres humanos no sos libres..
(defrule libertad_igualdad_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato SER_HUMANO ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe "ser" ?v))
(test (or (existe "libres" ?p) (existe "iguales" ?p)))
(test (or (existe "dignidad" ?p) (existe "derechos" ?p)))
=>
(bind ?existe_articulo (member$ 4 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 4))
```

```
))

;El hombre y la mujer, cualquiera que sea su estado civil, tienen iguales oportunidades y responsabilidades
(defrule estado_civil_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato PERSONA ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe "tener" ?v))
(test (existe "igual" ?p))
(test (or (existe "oportunidad" ?p) (existe "responsabilidad" ?p)))
=>
(bind ?existe_articulo (member$ 4 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 4))
))



;Los seres humanos no deben guardar conducta fraternal entre si
(defrule conducta_fraternal_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(adverbio ?a)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe_dato SER_HUMANO ?s))
(test (existe "deben" ?v))
(test (existe "guardar" ?a))
(test (existe "conducta" ?p))
(test (existe "fraternal" ?p))
=>
```

```
(bind ?existe_articulo (member$ 4 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 4))
))

;Toda persona tiene derecho a hacer lo que la ley prohibe
(defrule ley_prohibe_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado ?pn))
(test (existe_dato PERSONA ?s))
(test (existe "tener" ?v))
(test (existe_dato "derecho" ?p))
(test (existe_dato "ley" ?p))
(test (existe_dato "prohíbe" ?p))
=>
(bind ?existe_articulo (member$ 5 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 5)))
)


;Ninguna persona puede ser detenida o presa, sino por causa de delito o falta y en virtud de orden librada con apego a la ley
;por autoridad judicial competente
(defrule detencion_legal_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional ?c)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?sn)(neg_sustantivo_predicado nil))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?sn)))
(test (existe_dato PERSONA ?s))
```

```
(test (existe "ser" ?v))
(test (existe_dato FALTO_LIBERTAD ?p))
(test (existe "sino" ?c))
(test (or (not (existe_dato DELITO ?p)) (not (existe_dato FALTA ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 6))
))
```

;Los detenidos deberan ser puestos a disposición de otro que no sea autoridad judicial competente en un plazo que no exceda de seis horas

```
(defrule detencion_legal_negacion1
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "ser" ?v))
(test (not (existe_dato AUTORIDAD_JUDICIAL ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 6))
))
```

;Los detenidos no deberan ser puestos a disposición de la autoridad judicial competente

```
(defrule detencion_legal_negacion2
```

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe_dato NEGACION ?nv))
(test (existe "ser" ?v))
(test (not (existe_dato AUTORIDAD_JUDICIAL ?p)))
=>
(bind ?existe_articulo (member$ 6 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 6))
))


;Todo detenido podra ser informado de sus derechos
(defrule detenido_derechos_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado $?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "poder" ?v))
(test (existe "ser" ?v))
(test (existe "informado" ?p))
(test (existe_dato DERECHO_DETENIDO ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 8)))

)
```

```
;El detenido no podrá  declarar  ante autoridad judicial competente
(defrule declara_autoridad_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe "detenido" ?s))
(test (existe "poder" ?v))
(test (existe "declarar" ?v))
(test (existe "ante" ?p))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe_dato AUTORIDAD_JUDICIAL ?p))
=>
(bind ?existe_articulo (member$ 8 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 8)))
)


;Las autoridades judiciales no son la únicas competentes para interrogar a los detenidos o presos.
;no son las autoridades judiciales las únicas competentes para interrogar a los detenidos o presos.
(defrule interrogatorio_detenidos_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(adverbio ?a)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto ?ns)(neg_sustantivo_predicado nil))
(test (existe_dato AUTORIDAD_JUDICIAL ?s))
(test (or (existe_dato NEGACION ?nv) (existe_dato NEGACION ?ns)))
(test (existe "ser" ?v))
(test (existe "unicas competentes" ?p))
(test (existe "interrogar" ?p))
(test (existe_dato FALTO_LIBERTAD ?p))
=>
```

```
(bind ?existe_articulo (member$ 9 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 9))
))


;El interrogatorio extrajudicial no carece de valor probatorio
(defrule interrogatorio_extrajudicial_negacion
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo ?nv)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "interrogatorio" ?s))
(test (existe "extrajudicial" ?s))
(test (existe_dato NEGACION ?nv))
(test (existe "carecer" ?v))
(test (existe "valor" ?p))
(test (existe "probatorio" ?p))
=>
(bind ?existe_articulo (member$ 9 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 9)))

)


;Las personas aprehendidas por la autoridad  podrán ser conducidas a lugares de detención, arresto o prisión
;diferentes a los que están legal y pùblicamente destinados al efecto
;sinonimos de aprehendida
(defrule lugar_detencion_negacion
```

```
(object (is-a ORACION)(sustantivo_sujeto $?s)(verbo_aux nil)(verbo $?v)(condicional nil)(sustantivo_predicado ?p)(neg_verbo nil)(neg_sustantivo_sujeto nil)(neg_sustantivo_predicado nil))
(test (existe "persona aprehendida" ?s))
(test (existe "poder" ?v))
(test (existe "conducidas" ?p))
(test (existe_dato LUGAR_DETENCION ?p))
(test (existe "diferentes" ?p))
(test (existe "destinados" ?p))
(test (existe "efecto" ?p))
=>
(bind ?existe_articulo (member$ 10 ?*lista_i*))
(if (eq ?existe_articulo FALSE)
then

  (bind ?*lista_i* (insert$ ?*lista_i* 1 10))
))
```

ANEXOS

Artículos de la Constitución de la Republica:

ARTICULO 1º. Protección a la persona. El estado de Guatemala se organiza para proteger a la persona y a la familia; su fin supremo es la realización del bien común.

ARTICULO 2º. Deberes del Estado. Es deber del Estado garantizarle a los habitantes de la República la vida, la libertad, la justicia, la seguridad, la paz y el desarrollo integral de la persona.

ARTICULO 3º. Derecho a la vida. El estado garantiza y protege la vida humana desde su concepción, así como la integridad y la seguridad de la persona.

ARTICULO 4º. Libertad e igualdad. En Guatemala todos los seres humanos son libres e iguales en dignidad y derechos. El hombre y la mujer, cualquiera que sea su estado civil, tienen iguales oportunidades y responsabilidades. Ninguna persona puede ser sometida a servidumbre ni a otra condición que menoscabe su dignidad. Lo seres humanos deben guardar conducta fraternal entre sí.

ARTICULO 6º. Detención Legal. Ninguna persona puede ser detenida o presa, sino por causa de delito o falta y en virtud de orden librada con apego a la ley por autoridad judicial competente. Se exceptúan los casos de flagrante delito o falta. Los detenidos deberán ser puestos a disposición de la autoridad judicial competente en un plazo que no exceda de seis horas, y no podrán quedar sujetos a ninguna otra autoridad.

El funcionario, o agente de la autoridad que infrinja lo dispuesto en este artículo será sancionado conforme a la Ley, y los tribunales, de oficio, iniciarán el proceso correspondiente.

ARTICULO 7º. Notificación de la causa de detención. Toda persona detenida deberá ser notificada inmediatamente, en forma verbal y por escrito, de la causa que motivó su detención, autoridad que la ordenó y lugar en el que permanecerá. La misma notificación deberá hacerse por el medio más rápido a la persona que el detenido designe y la autoridad será responsable de la efectividad de la notificación.

ARTICULO 8º. Derechos del detenido. Todo detenido deberá ser informado inmediatamente de sus derechos en forma que le sean comprensibles, especialmente que puede proveerse de un defensor, el cual podrá estar presente en todas las diligencias policiales y judiciales. El detenido no podrá ser obligado a declarar sino ante autoridad judicial competente.

ARTICULO 9º. Interrogatorio a detenidos o presos. Las autoridades judiciales son las únicas competentes para interrogar a los detenidos o presos. Esta diligencia deberá practicarse dentro de un plazo que no exceda de veinticuatro horas.

El interrogatorio extrajudicial carece de valor probatorio.

ARTICULO 10. Centro de detención legal. Las personas aprehendidas por la autoridad no podrán ser conducidas a lugares de detención, arresto o prisión diferentes a los que están legal y públicamente destinados al efecto. Los

centros de detención, arresto o prisión provisional, serán distintos a aquellos en que han de cumplirse las condenas.

Las autoridades y sus agentes, que violen lo dispuesto en el presente artículo, serán personalmente responsables.