

CURSO 2016-2017
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Práctica final:
Sistema experto sobre cuestiones legales y
software.

Carlos Manuel Sequí Sánchez
DNI: 20 48 69 26 K

22 de junio de 2017

Índice

1	Resumen del funcionamiento del sistema experto	3
2	Descripción del proceso seguido para el desarrollo.	3
2.1	Procedimiento seguido para obtener el conocimiento.	3
2.2	Procedimiento de validación y verificación del sistema.	8
3	Descripción del sistema desarrollado	8
3.1	Variables de entrada del problema	8
3.2	Variables de salida del problema	8
3.3	Conocimiento global del sistema: hechos y relaciones iniciales	9
3.4	Especificación de cada módulo	10
3.5	Estructura de funcionamiento del esquema de razonamiento del sistema	11
3.6	Representación de los hechos utilizados por el sistema	11
3.7	Reglas de cada uno de los módulos	12
4	Uso del sistema	14

1. Resumen del funcionamiento del sistema experto

La función a desempeñar por el sistema experto creado para la práctica final de la asignatura es básicamente la de informar a un usuario inexperto en el tema acerca de cuestiones legales (como el uso de cierto tipo de datos) y de software (como el uso de licencias) de la misma manera que un experto humano podría hacerlo, sin dar lugar a confusiones y, haciendo ver al usuario que lo utiliza, qué tipo de decisión es mejor a la hora de escoger una licencia para su software, mostrarle las compatibilidades entre las diferentes licencias software e informarle sobre la ley de protección de datos.

El funcionamiento del sistema propio es simple, mediante una serie de preguntas realizadas por el sistema, se consigue guiar al usuario hacia la construcción de la respuesta que busca. El propio usuario decide cuál de los 3 módulos ejecutar para ser ayudado mediante dichas preguntas (elección de licencia software, compatibilidades entre licencias o asesoramiento sobre ley de protección de datos).

2. Descripción del proceso seguido para el desarrollo.

2.1. Procedimiento seguido para obtener el conocimiento.

Para la obtención del conocimiento en el **módulo 1** he utilizado las transparencias de la asignatura con el fin de poder conocer el procedimiento de creación del árbol de conceptos mediante la rejilla de repertorio y así establecer los caminos correctos que guiarán al usuario a la elección de la licencia idónea para su software.

Además, con el propósito de reunir la suficiente información de todas las licencias, es decir, para obtener los valores de las características escogidas para la realización de la rejilla de repertorio, he consultado páginas de internet en las que se habla de manera extensa de cada una de las licencias, lo que me ha permitido poseer un amplio conocimiento de estas.

Muestro las licencias escogidas de las expuestas en el guión de la práctica, además de las características que he decidido medir de dichas licencias junto con los posibles valores que pueden tomar.

Licencias escogidas:

- Apache software (AS)
- BSD original (BSD)
- GNU public (GPL)
- GNU reducida (LGPL)
- Mozilla public (MPL)
- Reciprocal public (RP)

Características medidas:

- **Copyleft**
 - Sin copyleft - 0
 - Copyleft reducido - 1
 - Copyleft limitado - 2
 - Con copyleft - 3
- **Capacidad de licencias dobles**
 - No - 0
 - Si - 1
- **Compatibilidad con GNU**
 - Si - 1
 - No - 0
- **Certificado OSI**
 - Si - 1
 - No - 0
- **Posesión de patentes**
 - No - 0
 - Opcional - 1
 - Si - 2
- **Licencia libre**
 - No - 0
 - Si - 1

A continuación muestro las tablas obtenidas con la aplicación del método de rejilla de repertorio sobre las distintas características elegidas por mí mismo, lo que nos dará información para facilitarnos la obtención de preguntas guiadas a realizar al usuario con el fin de obtener su licencia idónea. En verde clarito indicamos la casilla de menor valor, la cual nos indica qué características combinar en las sucesivas iteraciones.

	Copyleft	Licencias dobles	Compat. GNU	Certif. OSI	Patentes	Libre
AS	0	0	0	1	2	1
BSD	0	0	0	0	1	1
GPL	3	0	1	1	2	1
MPL	2	1	0	1	2	1
LGPL	0	0	1	1	2	1
RP	3	0	0	1	0	0

Figura 2.1: Rejilla de repertorio

	AS	BSD	GPL	MPL	LGPL	RP
AS		2	4	3	1	6
BSD			6	3	3	6
GPL				3	3	4
MPL					4	5
LGPL						7
RP						

Figura 2.2: Rejilla de repertorio

	BSD	GPL	MPL	RP	AS-LGPL
BSD		6	3	6	2
GPL			3	4	3
MPL				5	3
RP					7
AS-LGPL					

Figura 2.3: Rejilla de repertorio

	GPL	MPL	RP	AS-LGPL-BSD
GPL		3	4	3
MPL			5	3
RP				6
AS-LGPL-BSD				

Figura 2.4: Rejilla de repertorio

	GPL	RP	AS-LGPL-BSD-MPL
GPL		4	3
RP			4
AS-LGPL-BSD-MPL			

Figura 2.5: Rejilla de repertorio

	RP	AS-LGPL-BSD-MPL-GPL
RP		4
AS-LGPL-BSD-MPL-GPL		

Figura 2.6: Rejilla de repertorio

La siguiente figura es el árbol de decisión creado para la implementación del primer módulo:

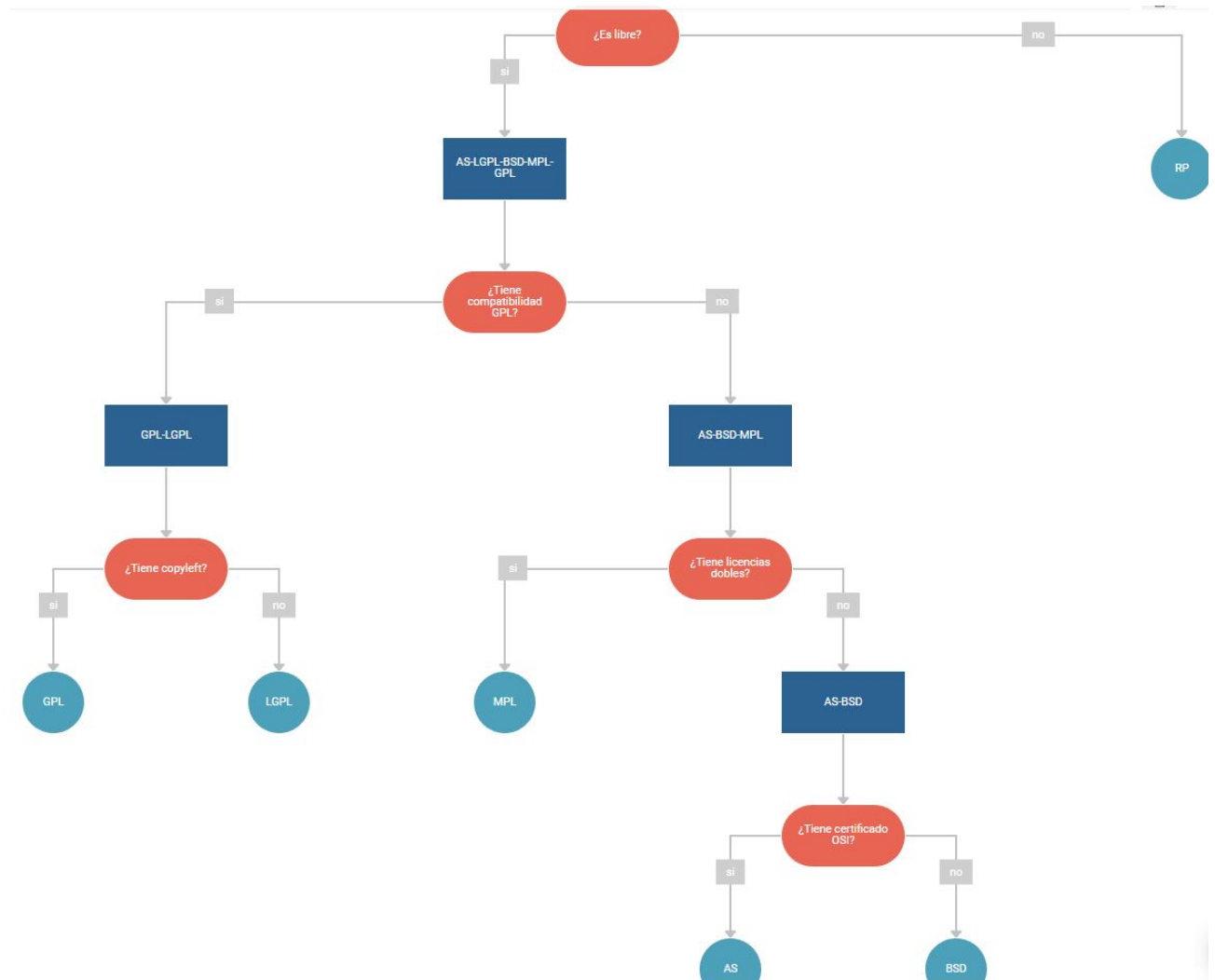


Figura 2.7: Árbol de decisión

Con respecto a la obtención de conocimiento en **el segundo de los módulos** que era necesario implementar, el proceso fue exactamente igual, es decir, la consulta de compatibilidades entre licencias software la realicé en las mismas páginas web. Sin embargo el árbol de decisión aprendido fue gracias al fichero CTree.xls proporcionado por los profesores de la asignatura. A continuación muestro el árbol de decisión obtenido modificando el fichero CTree.xls para que compare dos a dos todas las licencias, diciendo si son o no compatibles entre ellas y realice un profundo estudio mediante aprendizaje automático del árbol el cual nos contestará a la pregunta de "¿qué características he de comprobar para saber si dos licencias son compatibles?":

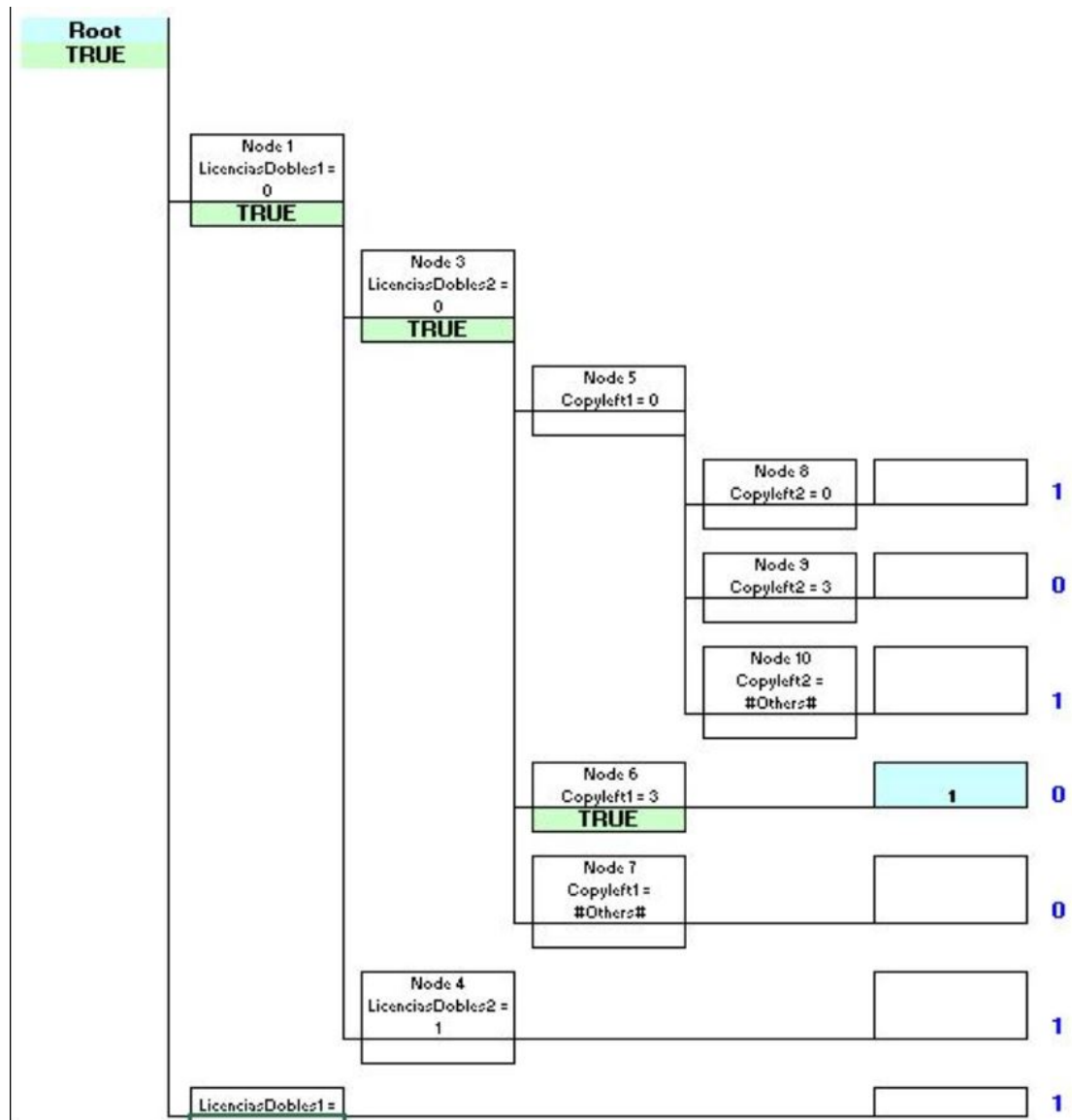


Figura 2.8: Árbol de decisión

Como podemos observar con tan solo dos características podemos averiguar si dos de las licencias seleccionadas por el usuario son compatibles entre sí, todo gracias al árbol de decisión aprendido, como he mencionado antes.

Con respecto al **módulo 3**, los datos manejados por el sistema (LOPD) han sido tomados de las páginas referenciadas en la guía de la práctica. El apartado 3 de dicho módulo ha sido implementado siguiendo mi propio criterio a la hora de decidir cuales son los datos que nos proporcionan una identificación total o parcial de un individuo.

2.2. Procedimiento de validación y verificación del sistema.

- **Verificación:** me he encargado de comprobar que el sistema no posee ningún tipo de error semántico, lógico o estructural, de tal manera que no sufre bloqueos en bucles, ni existen reglas que nunca puedan llegar a darse, al mismo tiempo que tampoco existen variables que puedan tomar valores ilegales o no válidos.
- **Validación:** he hecho las veces de usuario, ejecutando cada una de las casuísticas existentes en mi sistema y, comprobando la consistencia y robustez del mismo, al mismo tiempo que me he intentado asegurar de la certeza del conocimiento obtenido.

3. Descripción del sistema desarrollado

3.1. Variables de entrada del problema

Como variables de entrada de mi sistema distinguimos según el módulo que se ejecuta:

- **Módulo 1:** Las variables de entrada se corresponden con las sucesivas contestaciones que el usuario responde a las cuestiones que el sistema proporciona a la hora de encaminarlo por la ruta adecuada del árbol de decisiones.
- **Módulo 2:** En este módulo las únicas variables de entrada son las dos licencias para las cuales se desea saber si son compatibles con las características que posee cada una o no.
- **Módulo 3:** Cada una de las preguntas generadas por el sistema en este módulo hacen que el usuario tenga que introducir unas cuantas variables de entrada. Estas son por ejemplo, los datos a utilizar, el tipo de organización que los usará, el uso que se les dará...

3.2. Variables de salida del problema

Como variables de salida de mi sistema distinguimos según el módulo que se ejecuta:

- **Módulo 1:** En este módulo la variable de salida se corresponde con la licencia adecuada para el usuario en cuestión.
- **Módulo 2:** Aquí la variable de salida es una respuesta a: ¿es compatible la licencia A con la licencia B? Además de la explicación asociada a la respuesta.
- **Módulo 3:** Las variables de salida son dos en este caso. La primera es una clasificación de los datos que el usuario ha decidido utilizar. La segunda es la respuesta a la pregunta: ¿con los datos escogidos podemos identificar total, parcial o de forma nula a un individuo?

3.3. Conocimiento global del sistema: hechos y relaciones iniciales

A continuación comentaré los hechos y las relaciones creadas inicialmente para la ejecución de los distintos módulos.

■ Módulo 1

- tiposLicencias: establece para cada una de las características medidas, cuales son las licencias que la poseen.
- controlLicenciasEscogidas: nos permite mantener el control de lo que el usuario introduce por teclado y repetir las preguntas en caso de que se introduzca algo no válido.

■ Módulo 2

- característicasLicencias: lista de cada una de las licencias con cada una de las características que tienen para conocerlas en las reglas y poder comparar las unas con las otras a la hora de conocer si son compatibles o no.
- valoresCopyleft: lista de los posibles valores que puede tomar la características copyleft en cada una de las licencias. Puede tomar los valores: noCopyleft, copyleftLimitado y copyleft.
- valoresLicenciasDobles: exactamente igual que el hecho anterior pero, con la característica "licenciasDobles" pudiendo tomar los valores: sinLicenciasDobles y conLicenciasDobles.

■ Módulo 3

-
- identificacionDePersonas: hecho que ayuda en la resolución del apartado 3.3 de la práctica. En este hecho defino cuales (a mi parecer) son los datos que nos aportan una identificación total de un individuo, cuales son los que aportan una identificación parcial y cuales lo que aportan una identificación nula.
- tiposDatos: aquí definimos los "datos de carácter personal" proporcionados por la LOPD con cada uno de sus tipos asociados. Gracias a ello podemos realizar la impresión de los datos escogidos por el usuario y la clasificación de estos.
- tiposOrganizaciones: para el control de la introducción por teclado de los tipos de organizaciones que usarán los datos.
- tiposLugares: para el control de la introducción por teclado de los lugares posibles donde operarán los datos.
- tiposUsos: para el control de la introducción por teclado de los posibles usos que se le podrán dar a los datos.
- identificacionPersonas: deftemplate para asociar a cada dato el tipo de identificación que tiene.

3.4. Especificación de cada módulo

A continuación paso a describir cada uno de los módulos implementados:

- **Elección de licencia de software:**
 - **Objetivo:** Descubrir al usuario, de las licencias disponibles en el sistema, cual es la más adecuada en función de los requisitos que este desee para su software.
 - **Conocimiento utilizado:** El conocimiento usado es el obtenido mediante la rejilla de repertorio realizada a mano, la cual nos proporciona el árbol de decisión sobre el que guiar las ideas del usuario.
 - **Conocimiento deducido:** El conocimiento deducido son las posibles licencias que pueden obtenerse y el por qué de las elecciones tomadas en cada una de las casuísticas posibles dadas por el sistema experto.
- **Compatibilidades entre diferentes licencias software**
 - **Objetivo:** Definir compatibilidades entre parejas de licencias (la del software que se está utilizando y la que se desea utilizar). El objetivo es informar al usuario de dichas compatibilidades.
 - **Conocimiento utilizado:** El árbol de decisión creado, el cual nos conduce a saber cuales son las características que hemos de mirar para conocer de forma precisa si dos licencias de las que hay actualmente en la base de datos son compatibles o no.
 - **Conocimiento deducido:** La propia compatibilidad entre licencias software a partir de dicho árbol
- **Asesoramiento sobre ley de protección de datos**
 - **Objetivo:** Hacer conocer al usuario todo lo necesario acerca de la LOPD (Ley de protección de datos) para que este pueda actuar dentro del marco legal y los acuerdos preestablecidos, además de conocer los derechos ARCO a ejercer.
 - **Conocimiento utilizado:** Información extraída de las fuentes de información proporcionadas en el guión de la asignatura. Dicha información son tanto los datos de carácter personal a utilizar, como los tipos de organización que usarán los datos, como los usos posibles que se les dará a estos, etc.
 - **Conocimiento deducido:** En este caso el conocimiento deducido consiste en la clasificación de los tipos de datos que el usuario ha decidido utilizar en los distintos grupos de caracterización según la LOPD y, en segundo lugar, la inferencia proporcionada por el sistema de, si el conjunto de datos seleccionado por el usuario es suficiente como para identificar de forma total o parcial a un individuo.

3.5. Estructura de funcionamiento del esquema de razonamiento del sistema

Como ya he comentado, el funcionamiento del sistema está esquematizado de manera bien clara en tres bloques o módulos:

1. Elección de licencia software.
2. Consulta de compatibilidades entre licencias.
3. Asesoramiento sobre ley de protección de datos.

Cada uno de estos módulos posee una serie de hechos y reglas propios los cuales le permiten llevar a cabo el objetivo con el que se pretende ayudar al usuario. Para ejecutar o lanzar cada uno de los módulos, el usuario simplemente ha de elegir en el menú inicial creado para el sistema, cual de los tres bloques desea ejecutar, el 1, el 2 o el 3.

De esta manera, si ha introducido por teclado un 1, entonces el sistema lanzará el primer módulo con el fin de ayudar a dicho usuario en la elección de la licencia adecuada para su sistema software, y así sucesivamente para el resto de los módulos.

3.6. Representación de los hechos utilizados por el sistema

En este apartado me dispongo a formalizar la representación del conocimiento en cada uno de los hechos utilizados por el sistema

■ Módulo 1

- tiposLicencias: (<caracteristica><licencia1><licencia2>...)
- controlLicenciasEscogidas: (licenses <nombreLicencia1><nombreLicencia2>...), (licenciaEscogida nombreInicial). De esta manera cuando el usuario introduce un nombre por teclado, nombreInicial ha de ser igual a alguna de las licenses definidas.

■ Módulo 2

- característicasLicencias: (licencia <nombreLicencia><caracteristica1><caracteristica2>...)
- valoresCopyleft: (copy noCopyleft copyleftLimitado copyleft). (posibles valores que puede tomar copyLeft)
- valoresLicenciasDobles: (dobles sinLicenciasDobles conLicenciasDobles). (posibles valores que puede tomar licenciasDobles)

■ Módulo 3

- identificacionDePersonas: (dato_identificacion <nivelIdentificacion><dato>).
- tiposDatos: (tipoDato <tipo><dato>).
- tiposOrganizaciones: (tiposOrganizacionesPosibles UD EP ODAP), (tipoOrganizacionEscogido AAAA). De manera que cuando el usuario introduzca el tipo de organización por teclado, esta ha de coincidir con una de las definidas.

- **tiposLugares:** para el control de la introducción por teclado de los lugares posibles donde operarán los datos. (tiposLugaresPosibles asia europa oceania america africa), (lugarEscogido AAAA). De manera que cuando el usuario introduzca el lugar por teclado, este ha de coincidir con uno de los definidos.
- **tiposUsos:** (tiposUsosDisponibles domestico negocios judiciales seguridad), (usoEscogido AAAA). De manera que cuando el usuario introduzca el uso deseado por teclado, este ha de coincidir con uno de los definidos.
- **identificacionPersonas:** (field dato) (field tipoIdentificacion)

3.7. Reglas de cada uno de los módulos

Estas son las reglas utilizadas (explicadas de manera general) para la resolución de cada uno de los módulos. Todas las reglas tienen implícito un control de errores mediante el cual el sistema se asegura de que el usuario solo pueda introducir términos con valores permitidos, lo que garantiza el buen funcionamiento del sistema.

■ Módulo 1:

- **licenciasDisponibles:** Impresión por pantalla de las licencias que hay disponibles en el sistema.
- **preguntaX:** realiza la pregunta X del árbol de decisión aprendido mediante la rejilla de repertorio.
- **aceptarAfirmacionX:** en caso de que el usuario responda afirmativamente se lanza una regla A.
- **aceptarNegacionX:** en caso de que el usuario responda de forma negativa se lanza otra regla B.

■ Módulo 2:

- **lecturaLicenciaX:** realiza la lectura por teclado de la licencia X (donde X es 1 o 2, ya que solo se comparan dos licencias) que el usuario quiera escoger.
- **ramaArbolDecisionX:** son las reglas que controlan cada una de las ramas existentes en el árbol de decisión creado por CTree.xls. Hay 2 reglas menos que ramas existentes, ya que hay 2 ramas en el árbol creado que, con el conocimiento que posee mi sistema nunca llegarían a ejecutarse, por tanto hay 5 ramas en total. Cada una de las reglas controla en los antecedentes lo que ha de ocurrir para avanzar por dicha rama, controlando como he dicho antes, las únicas dos características que son necesarias para saber si dos licencias son compatibles en mi sistema: copyleft y licencias dobles.

■ Módulo 3:

- **mostrarDatosManejables:** muestra la lista de datos que son manejables por el sistema, los propuestos por la LOPD como "datos personales".

- **escogerDatos**: regla que permite al usuario escoger los datos que desee de la lista anterior.
- **denegarContestacion**: en caso de que se equivoque no aserta el hecho con lo que haya introducido el usuario.
- **aceptarContestacion**: en caso de que se introduzca bien el dato, se aserta.
- **finRecogidaDatos**: cuando el usuario escribe "fin" por el teclado, se dejan de tomar datos de teclado y se procede a imprimirlos con sus correspondientes tipos.
- **impresionTiposEscogidos**: imprimimos por pantalla los datos escogidos con los tipos a los que pertenecen.
- **tiposDeOrganizaciones**: primera pregunta del apartado 2 del módulo 3. Se muestran las posibles organizaciones y se pide al usuario que introduzca la que desee.
- **lugaresGeograficosDisponibles**: segunda pregunta del apartado 2 del módulo 3. Se muestran los posibles lugares geográficos y se pide al usuario que introduzca el que desee.
- **usosDisponiblesDeDatos**: tercera pregunta del apartado 2 del módulo 3. Se muestran los posibles usos de los datos y se pide al usuario que introduzca el que desee.
- **insertarTiposIdentificacionesADatos**: se encarga de asertar cada dato introducido por el usuario con el tipo de identificación que posee.
- **determinarIdentificacionPersonaX**: hay en total 5 reglas de este estilo, las cuales se encargan de analizar los tipos de identificación que pueden darse con los datos que ha escogido por teclado el usuario del sistema. Estos son los 5 tipos de identificaciones:
 1. Dos datos con tipo de identificación total, lo que nos da una identificación **total** del individuo.
 2. Dos datos con tipo de identificación parcial y uno con tipo de identificación **total**, lo que nos da una identificación total del individuo.
 3. Cuatro datos con tipo de identificación parcial, lo que nos da una identificación **total** del individuo.
 4. Tres datos con tipo de identificación parcial, lo que nos da una identificación **parcial** del individuo.
 5. Dos datos con tipo de identificación parcial, lo que nos da una identificación **parcial** del individuo.
- **determinarNoIdentificacionPersona**: esta regla se lanza única y exclusivamente cuando no se lanza ninguna de las anteriores, lo cual ocurrirá cuando el usuario haya escogido tan solo datos con tipo de identificación nula o un solo dato con tipo de identificación parcial.

4. Uso del sistema

Para la ejecución del sistema simplemente hay que ejecutar los siguientes comandos en CLIPS:

1. **(clear)**
2. **(load p2.clp)**
3. **(reset)**
4. **(run)**

Una vez realizado esto se mostrará el menú anteriormente comentado en el cual se da a elegir por el lanzamiento de uno de los tres módulos distintos exigidos en la práctica. Para elegir uno de estos habrá que pulsar las teclas 1, 2 o 3 en función del bloque que queramos ejecutar.

Por motivos de falta de tiempo durante estos últimos días, me ha sido imposible avanzar en la práctica a partir del punto 3.4, es decir, no he podido realizar los puntos 4 y 5 del módulo 3 de la práctica.