

Memoria practica final

Desarrollo de un S.E

David Castro Salazar

75936864-H

22/6/2017

1º Resumen del sistema experto

Este sistema experto consta de tres módulos. El primer módulo trata de la de averiguar la licencia que se quiere usar a través de unas cuestiones sobre las características de las licencias. El segundo módulo busca responder si hay compatibilidad entre dos licencias pidiendo características de cada una de ellas. Y por último el tercer módulo trata de mostrar las medidas que hay que tomar para el uso de datos según la LOPD.

2º Descripción del proceso de desarrollo

2.1. Módulo 1 y 2.

Para el módulo uno y dos he tratado de aparte de Buscar las características en la tabla que teníamos de referencia buscar por internet versiones o alguna licencia que no saliera en la tabla, he usado las más importantes y he querido que haya diversidad. Para ello he montado una tabla para sacar las características de cada licencia.

	GPL v3	LGPL v3	AGPL v3	Apache license	MIT license	BSD original	IBM public
patent use	si	si	si	si	no	no	si
copyleft debil-fuerte	2	1	3	0	0	0	2
libre	si	si	si	si	si	no	si
compatible con GNU	si	si	si	Si	si	no	no

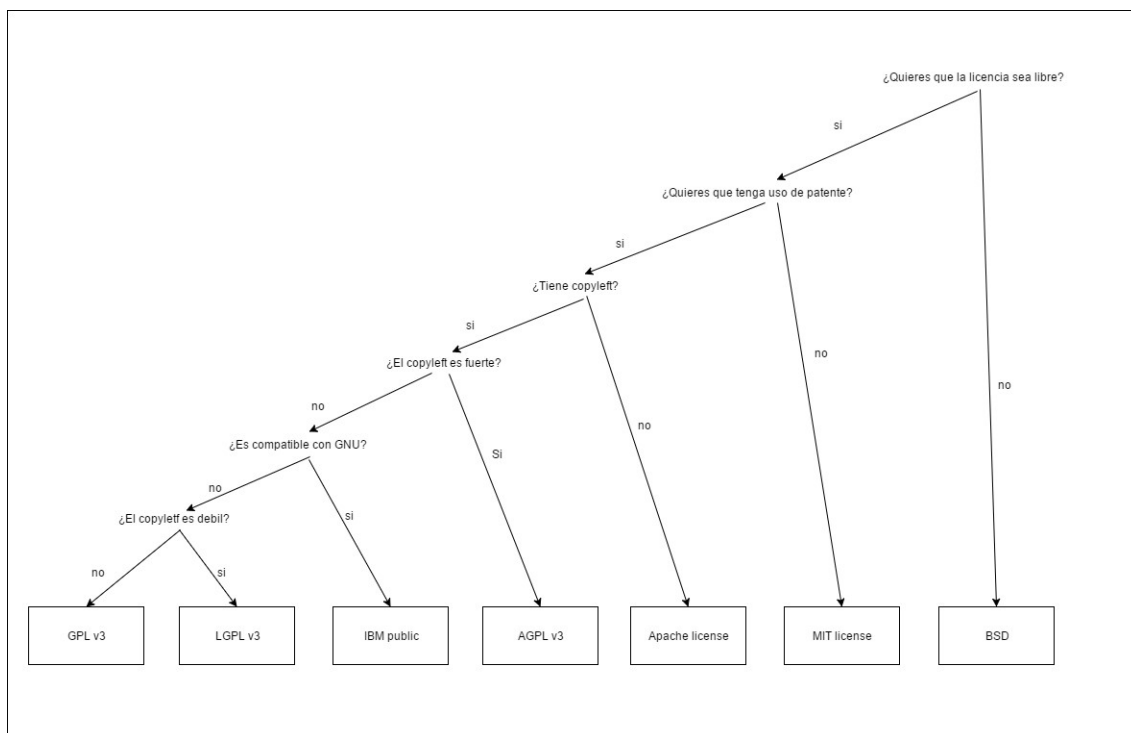
Una vez sacada las licencias y sus características (las que vamos a usar para diferenciarlas), hay que sacar otra tabla para ver cómo organizar las diferencias.

	GPL v3	LGPL v3	AGPL v3	Apache license	MIT license	BSD original	IBM public
patent use	1	1	1	1	0	0	1
copyleft debil-fuerte	2	1	3	0	0	0	2
libre	1	1	1	1	1	0	1
compatible con GNU	1	1	1	1	1	0	0
	5	4	6	3	2	0	4

Después de poner valor numérico a cada característica sacamos la tabla final con la que aplicaremos el método de la rejilla para sacar el árbol final.

	GPL v3	LGPL v3	AGPL v3	Apache license	MIT license	BSD original	IBM public
GPL v3		1	1	2	3	5	1
LGPL v3			2	1	2	4	0
AGPL v3				3	4	6	2
Apache license					1	3	1
MIT license						2	2
BSD							4
IBM public							

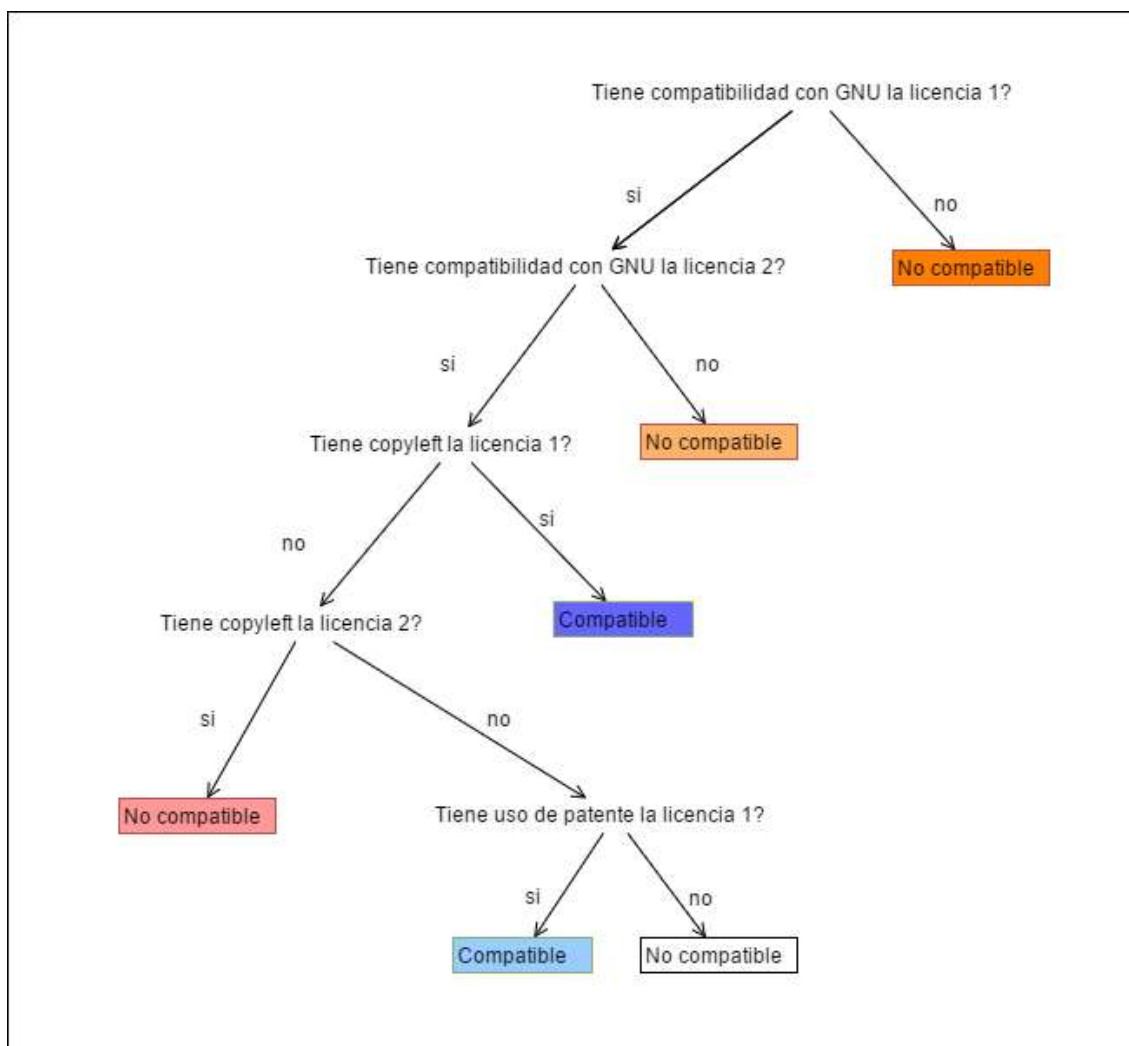
Esta es la tabla que quedaría, y a partir de ella sacamos el siguiente árbol para diferenciar cada licencia, y con ello poder conseguir sacar la licencia que se quiere en el modulo 1.



Para el modulo 2 he hecho otra tabla con las compatibilidades que hay entre las licencias (Esta búsqueda de compatibilidades no la he tenido tan clara, así que en la tabla puede haber alguna pareja de licencias mal puesta). Cada fila representa la licencia a la que se le va a introducir la otra.

	GPL v3	LGPL v3	AGPL v3	Apache license	MIT license	BSD original	IBM public
GPL v3		si	si	si	si	no	no
LGPL v3	si		si	si	si	no	no
AGPL v3	si	si		si	si	no	no
Apache license	no	no	no		si	no	no
MIT license	no	no	no	no		no	no
BSD original	no	no	no	no	no		no
IBM public	no	no	no	no	no	no	

Cada color de cada instancia representa en que pregunta esta cada compatibilidad, como se verá en el siguiente árbol.



De esta forma han sido obtenidos y tratados los datos de estos dos módulos.

2.2 Modulo 3.

Para el caso del modulo 3 no ha hecho mucha falta mirar paginas externas a las que nos han dado, Prácticamente de ahí se sacan los datos necesarios para realizar este modulo, la única dificultad ha sido obtener los datos que son necesarios para estar identificado, que al final los he puesto por opinión propia.

3º Descripción del Sistema de desarrollo

Para este SE he usado dos 3 template distintos, Uno para las licencias:

```
(deftemplate licencia
  (field Nombre)
  (field Uso-patente)
  (field Copyleftf)
  (field Libre)
  (field Compatibilidad-GNU)
)
```

Otro para los datos:

```
(deftemplate Dato
  (field Nombre)
  (field Tipo)
  (field Seleccionado)
)
```

Y otro para los aspectos:

```
(deftemplate Aspectos
  (field Organizacion)
  (field Zona)
  (field Uso)
)
```

Cada uno con las características necesarias para su funcionamiento.

Este SE comienza con la elección de sobre que modulo se ha a trabajar, un menú sencillo que te pide introducir un numero para el modulo al que se quiere ir.

```

(defrule opcion-Escogida
  (escoge_opcion ?opcion)
  (opcion ?opcion)
  ?v <- (volver empezar_de_nuevo) ;Para que volver ver a empezar no se ejecute reppetidas veces
  =>
  (if (eq ?opcion 1)
    then (assert (Eleccion-software1))
    else (if (eq ?opcion 2)
      then (assert (Compatibilidad))
      else (if (eq ?opcion 3)
        then (assert (proteccion))
        else (printout t crlf "Nos vemos." crlf)
      )
    )
  )
  (retract ?v)
)

```

Una vez escogida la opción el menú nos mandara a las reglas específicas. Para el caso del modulo uno nos manda a la regla,

```

(defrule Eleccion-software1
  (Eleccion-software1)
  =>
  (printout t "Responde si/no" crlf)
  (printout t "¿Quieres que la licencia sea libre?" crlf)
  (bind ?opcion (read))
  (if (eq ?opcion si)
    then (assert (Eleccion-software2))
    else (assert (Eleccion BSD))
  )
)

```

Esta regla tiene tres partes, en primer lugar el usuario tiene que responder a la pregunta que se le hace y depende de la respuesta se le llevara o a la siguiente pregunta con el hecho “(Eleccion-software2)”, o a la segunda “(Eleccion licencia)” que imprime la licencia que se ha escogido con sus características usando la regla,

```

(defrule Eleccion-licencia
  (Eleccion ?Nombre)
  (licencia
    (Nombre ?Nombre)
    (Uso-patente ?patente)
    (Copyleft ?copy)
    (Libre ?libre)
    (Compatibilidad-GNU ?compatibilidad)
  )
)

```

```

=>
(printout t crlf "La licencia que quiere es " ?Nombre ", esta licencia " ?patente
" tiene uso de patente, ademas su copyleft es " ?copy ", y " ?libre " es de software
libre, por ultimo " ?compatibilidad " tiene compatibilidad con GNU." crlf)
)

```

Con esto estaría descrito el modulo 1.

Para el modulo 2 nos mandaría a la regla,

```

(defrule Compatibilidad-software1
  (Compatibilidad)
  =>
  (printout t crlf "Responde si/no" crlf)
  (printout t "¿Tiene compatibilidad con GNU la licencia 1?" crlf)
  (bind ?opcion (read))
  (if (eq ?opcion no)
    then (printout t crlf "Las licencias no son compatibles" crlf)
    else (assert (Compatibilidad-software2))
  )
)

```

Este modulo es muy parecido al anterior ya que se basa en contestar preguntas y cada pregunta tiene dos casos, que te diga si es compatible o no, o que te lleve a la siguiente pregunta con el hecho "(Compatibilidad-softwareX)".

Por último el modulo 3 es más complejo ya que es el que más partes tiene, lo primero que hace es llevarnos a las reglas,

```

(defrule ImprimirDatos
  (proteccion)
  (Dato (Nombre ?nombre)
    (Seleccionado no)
  )
  =>
  (printout t ?nombre crlf)
)

```

```

(defrule Escoge-dato
  (proteccion)
  ?v <- (proteccion)
  =>
  (printout t crlf "Ponga el nombre del dato que vaya a usar. Si quiere dejar de meter datos pulse 0" crlf )
  (bind ?opcion (read))
  (printout t crlf crlf )
  (if (eq ?opcion 0)

```

```

        then (assert (clasificar1))
        else (assert (Seleccionar ?opcion))
    )
    (retract ?v)
)

```

Imprimirdatos como su nombre indica, muestra los datos que todavía no hayan sido seleccionados por el usuario. Y la segunda te hace escoger un dato para cambiar lo ha seleccionado, para saber sobre qué datos vamos a trabajar.

```

(defrule SeleccionarDato
  (Seleccionar ?dato)
  ?Dato <- (Dato (Nombre ?dato)
              (Seleccionado no))
  =>
  (modify ?Dato (Seleccionado si))

  (assert (proteccion))
)

```

Con esta regla cambiamos el dato ha seleccionado. Una vez escogidos los datos los mostramos ordenados por tipos de datos con las reglas,

```

(defrule Clasificar11
  (clasificar1)
  (Dato
    (Nombre ?nombre)
    (Tipo Protegidos)
    (Seleccionado si)
  )
  =>
  (printout t crlf " Datos protegidos:" crlf )
)

```

```

(defrule Clasificar12
  (clasificar1)
  (Dato
    (Nombre ?nombre)
    (Tipo Protegidos)
    (Seleccionado si)
  )
  =>
  (printout t ?nombre crlf )
)

```

Hay una regla para cada tipo de datos para que los muestre de forma ordenada. Una vez clasificados pasamos a pedir los aspectos necesarios con la regla,


```

(defrule Aspectos
  (solicitar-aspectos)

  ?Aspecto <- (Aspectos (Organizacion vacio))

  =>

  (printout t crlf "¿Que tipo de organizacion usara los datos?" crlf )
  (bind ?Organizacion (read))
  (modify ?Aspecto (Organizacion ?Organizacion))
)

(defrule Aspectos2
  (solicitar-aspectos)

  ?Aspecto <- (Aspectos (Zona vacio))

  =>

  (printout t crlf "¿En que zona geografica se van a almacenar los datos?" crlf )
  (bind ?zona (read))
  (modify ?Aspecto (Zona ?zona))
)

(defrule Aspectos3
  (solicitar-aspectos)

  ?Aspecto <- (Aspectos (Uso vacio))
  ?e <- (solicitar-aspectos)

  =>

  (printout t crlf "¿Que Uso se le dara a los datos?" crlf )
  (bind ?uso (read))
  (modify ?Aspecto (Uso ?uso))
  (assert (solicitar-aspectos-final ?uso))
  (retract ?e)
)

```

Con estas reglas ponemos cada aspecto que hay que rellenar. La siguiente parte es saber si con los datos que ha escogido el usuario se podría realizar una identificación de cada usuario (Esta parte la he hecho bajo mi opinión de que podría clasificar a cada persona) con la siguiente regla,

```

(defrule Identificacion
  (identificacion)
  ?e <- (identificacion)
  (Dato
    (Nombre DNI)

```

```

        (Seleccionado si)
    )
=>
(printout t crlf "Ya esta identificado con el DNI." crlf crlf crlf)
(assert (lopd))
(retract ?e)
)

(defrule Identificacion2
  (identificacion)
  ?e <- (identificacion)
  (Dato
    (Nombre ?nombre1)
    (Seleccionado si)
  )
  (Dato
    (Nombre ?nombre2)
    (Seleccionado si)
  )
  (Dato
    (Nombre ?nombre3)
    (Seleccionado si)
  )
=>

  (if (and (eq ?nombre1 Nombre_Apellidos) (eq ?nombre2 Direccion) (or (eq ?nombre3 Aficiones) (eq
?nombre3 Fecha_Nacimiento) (eq ?nombre3 Salud)))
    then (printout t crlf "Ya esta identificado con " ?nombre1 " , " ?nombre2 " y " ?nombre3 "." crlf )
    then (printout t crlf "Ya esta identificado con " ?nombre1 " , " ?nombre2 " y " ?nombre3 "." crlf crlf
crlf)
    (assert (lopd))
    (retract ?e)
  )
)

(defrule Identificacion3
  (identificacion)
  ?e <- (identificacion)
=>
(printout t crlf "Necesita mas datos para identificarse." crlf crlf crlf)
(assert (lopd))
(retract ?e)
)

```

He considerado que para estar identificado necesitas un DNI, en caso de no pedir el DNI se necesitara una tripleta de datos que tenga seguro el Nombre_Apellidos, la

Dirección y tenga aficiones, fecha_nacimiento o Salud. Están han sido las posibles opciones que he visto para registrarse y que se diferencia entre los usuarios.

Después de saber si estas identificado muestro las la LOPD para ver si se cumplen las normas para ello tengo una regla (muy larga) que es un menú que te muestra las normas según que quieras ver.

Por último he adjuntado un formulario PDF para el usuario pueda ejercer Iso derechas ARCO.

4º Manual del uso del sistema

Modulo 1 y 2.

Para el uso del modulo 1 y 2, tenemos que seleccionar primero el modulo y simplemente contestar las preguntas que nos va haciendo el SE.

Modulo 3.

El modulo 3 tiene cuatro partes. En primer lugar te pide seleccionar los datos que vas a usar introduciendo el nombre de cada dato. Una vez introducidos mostrara los nombres para ver que esta correcto. Acto seguido te pedirá que introduzcas una serie de aspectos necesarios. Después sacara si con los datos que has seleccionado tendrías suficiente para identificar a una persona. Y por ultimo saldrá un menú para poder ver las normas de la LOPD.