



**UNIVERSIDAD  
DE GRANADA**

# **INGENIERÍA DEL CONOCIMIENTO**

Práctica Final.

Desarrollo de un sistema Experto

*Lukas Häring García*  
47254273Z

<b>1. Resumen Sistema Experto.</b>	<b>2</b>
<b>3. Descripción del proceso seguido.</b>	<b>4</b>
3.1 Desarrollo de la base de conocimiento.	4
3.2 Procedimiento de validación y verificación.	5
<b>2. Descripción del sistema.</b>	<b>7</b>
4.a Variables de Entrada.	7
4.b Variables de Salida.	7
4.c. Hechos precargados.	8
4.d. Módulos que se han desarrollado.	8
4.e. Los hechos y las reglas de cada módulo.	9
<b>5. Uso del sistema.</b>	<b>13</b>
<b>6. Bibliografía</b>	<b>14</b>

# 1. Resumen Sistema Experto.

El sistema experto desarrollado en esta práctica es una hibridación de la práctica anterior donde era capaz de aconsejar una rama para un usuario.

En esta práctica, se han corregido los errores de la práctica anterior. El sistema experto será capaz de deducir qué rama según los gustos del usuario, sería más aconsejable para su selección, utilizando información del usuario como por ejemplo, los años en la carrera, la calificación media, gustos, etc...

En esta segunda parte, vamos a expandir la funcionalidad, donde al principio, el usuario podrá elegir entre ser aconsejado por ramas o aconsejado asignaturas.

Este último, añadirá razonamiento con incertidumbre, en concreto, se utilizará el razonamiento basado en factores de certeza, además permitirá al usuario como en el apartado anterior, la opción de "Siguiente", donde el sistema aplicará un valor por defecto para cada característica a la que se le ha aplicado la opción de "siguiente".

Una vez dentro de la opción de aconsejar asignaturas nos preguntará acerca del número de créditos a matricular, comprobando además que sean múltiplos de 8. Nos aparecerá de forma clara y ordenada mediante preguntas, el poder descartar o no el conjunto de asignaturas de un mismo año, en caso de no descartarlas, aparecerán una a una donde diremos "s(i) o n(o)" si queremos mantenerla o no, ofreciendo en todo momento al usuario información sobre la asignatura presente, sin tener que éste acordarse de los acrónimos.

Una vez elegidas las asignaturas deseadas, se nos pedirá información discreta sobre las preguntas, ya que como se ha comentado, se trata de un modelo por factores de certeza. Por ello, nos preguntará, por ejemplo, si nos gustan o no, las matemáticas, así lo preguntará con el software y con el hardware, finalmente una pregunta discretizada en 3 valores para la Nota, "Alta", "Media" o "Baja", su discretización será explicada más adelante.

Finalmente nos devolverá una lista de tantas asignaturas como créditos se haya informado inicialmente, estos estarán ordenados según el grado de certeza, cuanto mayor, más probable es de estar en dicha lista, aunque si dos tienen el mismo valor, estos serán añadidos según el orden en el que entraron como hechos (una pila).

### 3. Descripción del proceso seguido.

#### 3.1 Desarrollo de la base de conocimiento.

Para la obtención del conocimiento de la primera práctica, mi hermano realizó de extractor de conocimiento, mientras yo hacía de experto. Se realizó una entrevista en la que él iba ofreciéndome características personales y yo iba asesorando apoyado en la información que me daba y descartando con un motivo según la información recibida. Con toda esa información, se creó un grafo y se modeló sobre el sistema. (Véase figura 2.)

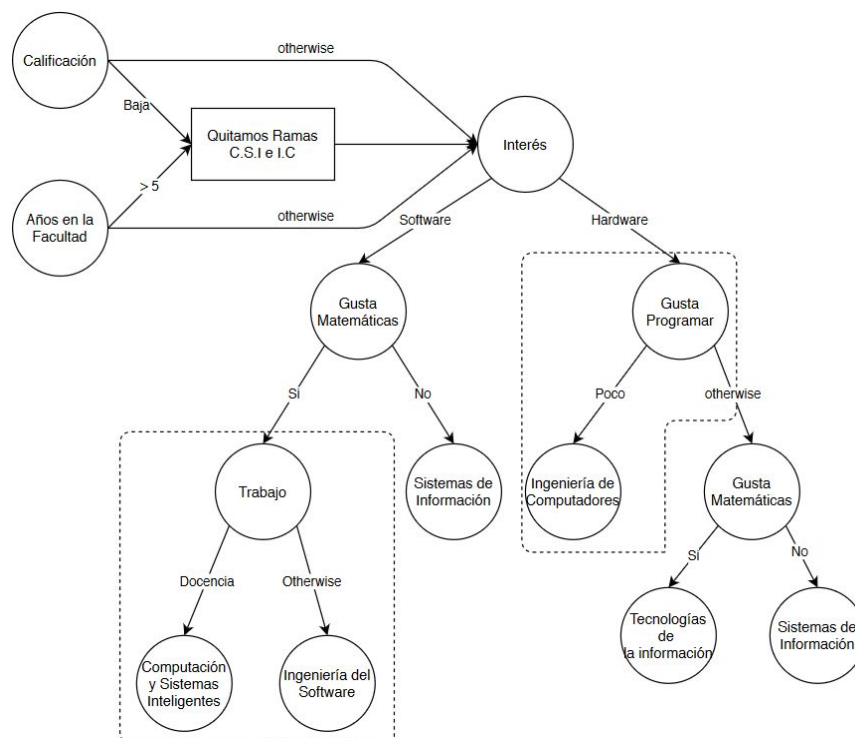


Figura 1. Grafo de representación práctica 1.

Para el segundo modelo que consiste en aconsejar unas asignaturas, de manera similar, inicialmente el usuario venía con un conjunto de asignaturas y se le informaba de un número de asignaturas a aconsejar (equivalente a créditos), mi compañero Miguel Lentisco hacía de experto e iba intuyendo a partir de mis respuestas cuál podía ser la asignatura más recomendable, y que evidencias eran destacables para la selección de una asignatura, por ejemplo, para una asignatura de matemáticas, por ejemplo cálculo, debe haber una evidencia de gustos hacia las matemáticas o que debe gustarte el hardware para Estructura de Computadores.

### 3.2 Procedimiento de validación y verificación.

Para la práctica inicial, vamos a generar una serie de casos pruebas, estos casos han sido generados por mí, donde Miguel Lentisco será nuestro Experto.

En todo momento se comprueba que la sintaxis es correcta, ya que es un requisito indispensable.

Casos	Dificultad	Respuesta Exp.	Respuesta Sist.
Llevo 4 años en la carrera, con una nota media de 8.0, me gusta el SW, las matemáticas y me gusta la docencia.	Fácil	CSI	CSI
Llevo 5 años en la carrera, mi nota es de un 5.2, me gusta el Hardware, me da igual programar, pero no me gustan las matemáticas	Media	SI	SI
Llevo 6 años en la carrera, mi nota es de un 6, me gusta el Software, me gustaría trabajar en la docencia.	Difícil	SI	SI

El sistema ha sido capaz de resolver la pregunta difícil ya que se implementa un método de descarte, cogiendo el último hecho no eliminado, aunque no tiene por qué haber salido correcta ya que este valor es obtenido debido al orden en el que se han introducido los datos.

Se considera más difícil entre ellos debido al número de descartes que ocurren o información incompleta (que incrementa la dificultad). Ya que en el último, no se especifica "Gusta matemáticas", pero el experto ha elegido esa opción debido a que su nota y los años en la carrera aconsejan que este haga la rama de SI.

Al tratarse de un sistema de grafo, podemos comprobar de forma sencilla que es completo ya que tiene un nodo hoja como solución a cada camino a tomar, preciso ya que da información de cada uno de las actuaciones realizadas. Distintos expertos pueden dar distintos resultados, por ejemplo, se podría haber recomendado la rama de Software. Consideraríamos las distintas vistas por los expertos como ciertos. Es por ello que no hay ninguna regla inalcanzable, ni inconsistente.

Para ver la validación del sistema al completo, tenemos que analizar también la otra parte, la recomendación de asignaturas, de manera similar al anterior, vamos a proponer 3 casos incrementando la complejidad. Las de color rojo serán añadidas en la dificultad media

Casos	Dificultad	Respuesta Exp.	Respuesta Sist.
<b>(MP, TOC, FFT, ES, CA, ED) 16 Créditos</b> No me gustan las matemáticas me gusta el software y me gusta el hardware, nota media un 6.0.	Fácil	MP, TOC	MP, TOC
<b>(FP, MP, FFT, ES, CA, ED) 16 Créditos</b> Solo me gusta el software y mi nota media es un 7.0	Media	FP, MP	FP, FFT
<b>(ALEM, IG, CA, FP, MP, ED, FFT) 24 Créditos</b> Me gusta el matemáticas y el hardware, mi nota media es un 9.0	Difícil	ALEM, CA, TOC	ALEM, CA, TOC

Vemos que el sistema siempre ofrece una solución, aunque claro, si el usuario muestra da menos asignaturas que créditos, se asignan todos por defecto, por el contrario, elige tantos como créditos sean necesarios basados en el valor de certeza.

Vemos que la consistencia es clara ya que siempre ofrece una solución, debido a la naturaleza del modelo de certeza. Por la misma naturaleza, el experto puede dar diferentes soluciones ya que no es un modelo perfecto, de ahí la palabra “certeza”. Los resultados obtenidos son buenos, se corresponde con valores cercanos al del experto.

No existe ninguna regla inalcanzable ya que se tratan de reglas que son aplicadas según las evidencias estrictamente un resultado, ya que si tuviéramos una regla (No A y A), esto sería inconsistente, de ahí recalcar que **solo una**.

Errores en sintaxis no hay ya que son respuestas simples y donde se comprueba que si existe un error, entonces se vuelve a preguntar.

Podemos concluir que nuestro sistema es consistente y válido.

## 2. Descripción del sistema.

Se encuentra modulado en 3 archivos, un archivo genérico que llama a los dos otros archivos o módulos.

El archivo "main" es "Sistema\_Experto\_Lukas\_Haring\_Garcia.clp", mientras que los otros dos módulos son llamados desde este y reciben el nombre de "Recomendador\_Asignatura.clp" y "Recomendador\_Rama.clp", estos están definidos con la regla "defmodule" para acoplarse a un módulo.

### 4.a Variables de Entrada.

Para la primera parte de la práctica las variables de entrada son la fecha de inicio, que se trata de un entero como la calificación, que se trata de un número restringido entre 0 y 10. Una pregunta binaria acerca del gusto o no de las matemáticas, un multichoice, que da la opción entre Docencia, Empresa o Autónomo, que refleja el futuro del usuario. Un rango de 0 al 10 que es discretizado para el gusto por la programación, donde [0-4] representa poco; (4-7] representa un gusto medio y (7-10] representa el gran gusto a la programación. Existe otra variable muerta que se utiliza solo para el nombre y no es utilizada nunca más. Todos estos son representados utilizando un hecho "(NombreVariable ?Valor)", por ejemplo "(Calificacion\_Numerica 10)"

Para la segunda parte nos encontramos con varias variables discretas, 3 variables binarias Si/No que pregunta sobre el gusto por las matemáticas, el gusto por el hardware y por el software, finalmente una discretización del valor que toma una nota del 0-10, donde [0,5] es una nota "Baja", (5, 7] nota "Media" y (7, 10] una nota "Alta". De la misma forma que la representación anterior, la estructura que utilizamos es "(NombreVariable ?Valor)", por ejemplo, "(NumCreditos 16)".

### 4.b Variables de Salida.

La variable de salida en el primer apartado es una cadena que indica la rama recomendada y el motivo por el cual se eligió esa rama, representado con un hecho "(Consejo ?Rama ?Texto ?Nombre)", nombre es utilizado para darle un nombre al sistema y crear mayor cercanía con el usuario.

Mientras la recomendación en la asignatura se trata de una “lista” de hechos con mayor valor de certeza, la salida es una cadena donde comenta la asignatura y la confianza a la que la recomienda. El hecho tiene la siguiente estructura “(FactorCerteza ?Hecho ?Si/No ?valor\_Certeza)”.

#### 4.c. Hechos precargados.

Los hechos precargados en la primera parte son las diferentes ramas que hay en el grado de ingeniería informática, cada hecho añadido tiene la siguiente estructura: “(Rama NombreRama)”, que se encuentran englobadas por un **deffacts**.

De manera similar para la segunda parte, añadimos hechos referentes a las **Asignaturas** donde la estructura es algo más compleja ya que ayuda a que el sistema se pueda generalizar. El hecho que se añade es:

“(Asignatura ?Nombre\_Asignatura ?Abreviatura ?Curso)”, por ello si queremos añadir un nuevo curso, solo basta con incrementar  $?Curso = Total\ Cursos + 1$  y el sistema verá que existen más cursos para preguntar sin la necesidad de añadir ninguna regla. Cabe destacar que para no complicar mucho el problema, se han implementado 25 asignaturas, primero, segundo y primer cuatrimestre de 3.

En el módulo “Main”, se utiliza una variable global que indica el path absoluto para la carga de los módulos, es importante modificarlo según el sistema ya que si no, no cargaría los módulos.

#### 4.d. Módulos que se han desarrollado.

Se han desarrollado **dos** módulos, como se ha especificado antes, un módulo para el desarrollo de la parte A), con el nombre “Recomendador\_Rama” en el archivo “Recomendador\_Rama.clp” en él contiene todo y cada una de las funciones, reglas, necesarias para el correcto funcionamiento de la práctica.

Mientras que el segundo módulo “Recomendador\_Asignatura.clp”, todas las reglas y funciones para aplicar los factores de certeza así como los elementos propios, este módulo recibe el nombre de “Recomendador\_Asignaturas”.

Estos archivos son cargados sobre “Sistema\_Experto\_Lukas\_Haring\_Garcia.clp” utilizando la función “load”, una vez cargados, según la selección, podemos cargar su contexto utilizando el método “(focus ?Nombre Modulo)”.



#### 4.e. Los hechos y las reglas de cada módulo.

Para la realización de esta parte, voy a resumir ya que si no, podría escribir 200 páginas.

Regla	Descripción
Pregunta_Nombre	Realiza una pregunta y una lectura, se utiliza para presentación, también pregunta sobre el año de iniciación de la carrera, que es una variable de entrada y será útil para descartar.
Saluda_Analiza	Comprobamos que el nombre tiene al menos tres caracteres, saludamos y pedimos información sobre la calificación media.
MuchosAniosCursados	Si la cantidad de años cursados es mayor que 5 significa que te está costando la carrera y proponemos a descartar dos ramas.
Elimina_Rama_Y_Marca	Elimina el hecho con la rama propuesta a borrar y la propuesta.
Elimina_Marca	Elimina solo la propuesta.
Aniado_Propuesta	Comprueba que solo ha quedado una rama a aconsejar, si es así añade la propuesta final.
Pide_Calificacion	Pide la calificación numérica (aunque no se puede deducir al principio).
Analiza_Nota	Comprueba que la nota esté entre 0 y 10, si la nota es inferior a 5, discretiza la nota como se ha comentado, si es inferior a 5, quitamos CSI e IC, añadimos la variable ?Nota_Val y pedimos información sobre el interés.
Pide_Gustos	Pregunta al usuario si su interés es sobre hardware o software, si es hardware, bifurcándose en dos caminos, añade una nueva variable.
Pide_<Informacion>	Comprueba cada casuística según la información a pedir y añade un hecho, que es una variable de entrada según la información ofrecida por el usuario.

Regla	Descripción
Interes_Software	Es uno de los caminos de la bifurcación, elimina aquellas ramas que no sean de software, por ejemplo IC y TIC. Dando una explicación.
Interesa_Software_Gusta_Matematicas	Si estamos por la ramas de software e interesa matemáticas, eliminamos la rama S.I ya que requiere pocas matemáticas. Dando una explicación.
Interesa_Software_Gusta_Matematicas_Docencia	Si me gustan las matemáticas y estoy en la ramas ramas de software y quiere dedicarse a la docencia, descartamos I.S y damos una explicación.
Interesa_Software_Gusta_Matematicas_No_Docencia	De manera similar a lo anterior, pero esta vez no es docencia, y eliminamos la rama CSI y damos una explicación.
Interesa_Software_No_Gusta_Matematicas	Si no gustara matemáticas, directamente eliminamos dos casos, CSI e IS. Dando un motivo.
Interesa_Hardware	Es otro de los caminos de la bifurcación, elimina aquellas ramas que no sean de hardware, por ejemplo CSI e IS. Dando una explicación.
Interesa_Hardware_Programar_Poco	Si le gusta el hardware y programar poco, eliminamos las dos ramas de TSI e SI, dando una explicación.
Interesa_Hardware_Programar_Mas_Poco	Si le gusta el hardware y programar medio o superior, eliminamos las IC, dando una explicación.
Interesa_Hardware_Programar_Mas_Poco_No_Matematicas	Si le gusta el hardware y programar medio o superior, pero no las matemáticas, eliminamos las T.I, dando una explicación.
Interesa_Hardware_Programar_Mas_Poco_Si_Matematicas	Si le gusta el hardware y programar medio o superior y también las matemáticas, eliminamos las S.I, dando una explicación.
Recomienda_<Rama>	Comprueba que se recomienda la rama y da información.
HayConsejo	Prioridad máxima, si hay un consejo, lo muestra.

De la misma forma que en el apartado anterior, se pretende resumir ya que mucho del código es repetir reglas que son modificadas levemente.

Regla	Descripción
certeza_evidencias, encadenado, combinacion	Se encuentran definidos en las diapositivas de
r<N>	Representa una regla de certeza que se define para aumentar o disminuir el valor de certeza de cada una de las asignaturas, utiliza las reglas antes descritas como se puede ver en la diapositiva del Tema.
Main_Recomendador_Asignatura	Estado inicial y declara la pregunta sobre el número de créditos.
Pregunta_Creditos	Preguntamos sobre los créditos necesarios y leemos la entrada como una variable NumCreditos.
Numero_Creditos	Comprueba que dicha variable es mayor de cero y es divisor de 8, si es así, crea unas variables que mantienen la información del total de créditos adjudicados, un contador para el curso a proponer para descartar.
Error<Nombre>	Comprueba que la variable no ha sido aceptada y si es así, indica un mensaje de error y vuelve a realizar la pregunta.
Seleccionar_Curso	Con el contador de curso comentado anteriormente, lo utilizamos para buscar al menos la existencia de una asignatura, si es así, entonces pregunta si quiere eliminarse todo el curso o no. Si se elige que no, toda las asignaturas quedan eliminadas, si se dice que si, irá asignatura a asignatura.
Elimina_Curso_Entero	Elimina el curso entero
Termina_Elimina_Curso	Cuando se elimina el curso entero, incrementa el contador y elimina el marcador de borrado.
Descartar_Asignaturas	Coge la primera asignatura del curso actual y pregunta si quiere o no ser escogido, si se escoge, se crea un nuevo hecho "(SelAsignatura)" y se incrementa un contador. Se elimina la asignatura.

Regla	Descripción
Elimina_Preguntadas	Pasamos al siguiente curso si ya no quedan elementos a seleccionar del curso actual y eliminamos su marca.
Seleccionar_Asignaturas_Final	Si ya no hay más asignaturas, avanzamos con las preguntas, primero, gusto por las matemáticas.
Preguntar_<Nombre>	Realiza una pregunta Si/No sobre si le gusta o no las <Nombre>, crea una respuesta que es utilizada para ser analizada. Si fuera error, iría a Error<Nombre>(Error_Recomendado_Matematicas)
Recomendado_<Nombre>	Vemos si la respuesta es de tipo binaria y si es así, la guardamos, se guardará como una Evidencia
Default_<Nombre>	Si se escribe siguiente, se añade el valor por defecto definido por el usuario, por ejemplo, para las preguntas binarias, se asignaría 'si', para la calificación un 'Baja'.
Preguntar_NotaMedia	Este es similar a los anteriores, pero se va a comentar, se crea una discretización de valores donde [0,6] es 'Baja'; (6, 8] 'Media' y (8, 10] 'Alta', por lo demás es igual a Preguntar_<Nombre>.
Resolver	Cuando se ha realizado la última pregunta, se busca aquel FactorCerteza que sea máximo y además, debe ser una asignatura seleccionada, además comprobamos que no hemos superado el límite de créditos asignados, para cada elemento incrementamos en 8 los créditos asignados, eliminamos dicho factor de certeza que lo identificaba y mostramos un mensaje sobre la información de la asignatura escogida.

Ejemplo de una restricción, por ejemplo "r3":

```
(FactorCerteza gusta_matematicas si ?f1)
(FactorCerteza gusta_hardware si ?f2)
(test (and (> ?f1 0) (> ?f2 0)))

=>
(assert (FactorCerteza FFT si (encadenado (* ?f1 ?f2) 0.8)))
```

Para el archivo main, tenemos las siguientes reglas

Regla	Descripción
Main	Se encarga de cargar los archivos que contienen los módulos, seguidamente de abrir el menú.
Menu	Muestra una cadena que contiene la información visual del menú. se realiza una lectura del teclado en una cadena larga y se guarda la variable de respuesta que será analizada para ejecutar uno u otro módulo.
Asesoramiento_Rama	Si contiene las palabras clave “asesoramiento” y “rama”, en el mismo orden, se cambia de vista al módulo recomendador de ramas.
Asesoramiento_Asignaturas	Si contiene las palabras clave “asesoramiento” y “creditos”, en el mismo orden, se cambia de vista al módulo recomendador de asignaturas.
Some_Other_Response	Si se responde algo no válido, se vuelve a mostrar el menú.

## 5. Uso del sistema.

Como se ha comentado, en el archivo “Sistema\_Experto\_Lukas\_Haring\_Garcia.clp”, se modificará la variable global para coincidir con nuestro path absoluto.

Ejecutamos todo el proyecto únicamente llamando al “main”.

```
(clear)
(load "<PATH>/Sistema_Experto_Lukas_Haring_Garcia.clp")
(reset)
(run)
```

Una vez ejecutado, utilizaremos únicamente nuestro teclado, la primera vista será un menú el cual podemos elegir una de las dos opciones, se puede realizar de diferentes formas siempre y cuando mantenga las dos palabras claves anteriores.

Una vez hemos elegido, nos hará una serie de preguntas y responderemos, el sistema **acaba cuando nos ha mostrado el resultado.**

## 6. Bibliografía

1. *Grau en Enginyeria en Informàtica*  
<https://www.cs.upc.edu/~bejar/ia/material/laboratorio/clips/FAQ-CLIPS.pdf>
2. CLIPS User's Guide:  
<https://www.csee.umbc.edu/portal/clips/usersguide/ug6.html>
3. Diapositivas de clase