

Sistema Experto

Practica Final - IC

Miguel Lentisco Ballesteros

77249107L

Índice

1. Resumen del funcionamiento del sistema experto	2
2. Descripción del proceso seguido	3
a. Procedimiento seguido para el desarrollo de la base de conocimiento	3
b. Procedimiento de validación y verificación del sistema	4
3. Descripción del sistema	5
Subsistema Menu Principal	5
Subsistema Recomendar Rama	5
Subsistema Recomendar Asignaturas	6
a. Variables de entrada del problema	7
Subsistema Menu Principal	8
Subsistema Recomendar Rama	8
Subsistema Recomendar Asignaturas	8
b. Variables de salida del problema	9
Subsistema Recomendar Rama	9
Subsistema Recomendar Asignaturas	9
c. Conocimiento global del sistema	9
Subsistema Menu Principal	10
Subsistema Recomendar Rama	10
Subsistema Recomendar Asignaturas	10
d. Especificación de los módulos	14
i. Estructura de módulos	14
ii. Descripción del módulo	14
e. Hechos y reglas	15
Subsistema Menú Principal	15
Subsistema Recomendar Rama	16
Subsistema Recomendar Asignaturas	18
4. Breve manual de uso del sistema	23

1. Resumen del funcionamiento del sistema experto

Se nos pide que implementemos un sistema experto que permita recomendar a los estudiantes de Ingeniería Informática que rama (mención) escoger, y también recomendar asignaturas para matricularse. Para hacer estas recomendaciones el sistema extrae información del usuario mediante preguntas como su nota media o intereses por temas junto a la información de las asignaturas (conocimiento de la base de datos), que procesará debidamente según el tipo de recomendación y dará una respuesta al usuario indicando las razones de su recomendación.

El sistema empieza con un menú principal que nos permite elegir los dos tipos de recomendaciones (mención/asignaturas):

- En el caso de la mención se empiezan a hacer una serie de preguntas (intereses, nota media) donde puede responderse o no, incluso parar en cualquier momento. El sistema evalúa la información recibida y da una o varias ramas posibles según las respuestas del usuario.
- Para las asignaturas, se muestra otro menú donde podemos ver la lista de asignaturas que tiene la base de datos o pasar al recomendador de asignaturas. Este hará distintas preguntas (nº créditos a matricular, lista de asignaturas para escoger, mención, intereses...) donde excepto las asignaturas y los créditos, el resto de preguntas son saltables, de manera que usamos Razonamiento por Defecto para lidiar con el problema de la información completa; se realizan distintos cálculos y el sistema muestra las asignaturas que debería cursar el estudiante junto a la justificación del sistema.

En ambos casos se vuelve al menú anterior para poder realizar otra recomendación si se desea, y también se puede salir del programa cuando el usuario termine.

Además el sistema está dividido en 3 archivos por las distintas funcionalidades (menú principal y recomendadores) y en cada archivo se subdivide en módulos (menú, hacer preguntas, cálculo, mostrar recomendación), de manera que separamos los distintos procesos para que sea mucho fácil entender y modificar el sistema, añadiendo también más fiabilidad al sistema.

2. Descripción del proceso seguido

a. Procedimiento seguido para el desarrollo de la base de conocimiento

Para desarrollar la base del conocimiento primero necesitamos estudiar como obtener el conocimiento necesario para efectuar buenas recomendaciones, inicialmente se partió estudiando el problema de recomendar mención. En concreto hicimos los siguientes estudios:

- Entrevista a un alumno de informática (ejercicio de clase): se entrevistó al experto Lukas Häring para obtener distintas valoraciones sobre los aspectos más fundamentales a la hora de escoger rama (por ej gustos, nota media. . .).
- Rejilla de repertorio (ejercicio de clase): usando estos aspectos, recopilamos de distintos estudiantes sus valores correspondientes junto a la mención deseada, agrupando la información por rama.
- Árbol con reglas (ejercicio de clase): usando la información recopilada, creamos un árbol de decisión que traduciendo a reglas nos da un algoritmo para recomendar la mención.

Valorando toda la información extraída, se consideró que lo más fundamental a la hora de escoger una mención era obtener la información sobre los gustos de temas relacionados con cada mención junto con la nota media del estudiante.

Después pasamos a obtener el conocimiento para recomendar asignaturas, reutilizando el conocimiento extraído anterior y añadiendo más de estos estudios adicionales:

- Otra entrevista con alumno de informática: se volvió a hacer una entrevista al experto Lukas Häring para obtener más características importantes a la hora de decidir sobre un conjunto de asignaturas.
- Ontología (ejercicio de clase): también se utilizó una representación con ontología de los distintos elementos que pueden actuar a la hora de escoger asignaturas.

Valorando de nuevo el conocimiento extraído, concluimos que hay múltiples factores que pueden afectar a la hora de coger una asignatura, tales como: si es obligatoria u optativa, si hay de distintos cursos, según la rama de la asignatura, los temas relacionados, su carga de trabajo y dificultad; la mención del usuario, sus ganas de trabajar, nota media. . .

Por tanto hemos seguido un proceso de estudio que nos ha permitido obtener las características y factores más relevantes que necesitamos obtener (el conocimiento que queremos buscar) para poder crear un sistema que resuelve correctamente nuestro problema.

b. Procedimiento de validación y verificación del sistema

Para asegurarnos que el sistema es de calidad se ha realizado un trabajo de validación y verificación, ya que buscamos que el sistema resuelva nuestro problema, su funcionamiento sea confiable, explique las razones de las recomendaciones y nos permita añadir conocimiento fácilmente.

Para el proceso de verificación se ha buscado cumplir los siguientes criterios, de manera que se han realizado pruebas y comprobaciones para cumplirlos mientras se construía el sistema:

- **Consistencia:** hemos procurado evitar conclusiones incoherentes, tanto de tipo estructural (reglas inalcanzables), lógicas (conclusiones o antecedentes redundantes, subsunción de reglas o reglas que producen contradicciones lógicas) o semánticas (valores ilegales en variables).
- **Precisión:** hemos procurado corregir errores de sintaxis y morfológicos, de manera que todo el sistema está bien escrito.
- **Completitud:** hemos considerado siempre todos los casos posibles, es decir, para todas las configuraciones posibles de respuesta del usuario siempre habrá una respuesta acorde.

Para el proceso de validación, hemos buscado que el sistema cumpla las especificaciones del modelo de diseño y satisfaga las necesidades del usuario, en concreto:

- La representación y técnica de razonamiento elegidas son correctas ya que el conocimiento está representado reflejando el modelo conceptual, siendo muy fácil de añadir nuevo conocimiento o modificarlo, dando además las respuestas con razones sólidas.
- El diseño y la implementación está completamente modulado de manera que dividimos las distintas tareas del sistema permitiendo una buena comunicación entre subsistemas, y haciendo que el sistema sea fácil de comprender, modificar y mantener.
- El sistema responde en tiempo real y su interfaz sencilla permite que el usuario no tenga ningún problema en usar el sistema.

Finalmente se han realizado múltiples casos de prueba (normales, extremos) para comprobar que todo, tanto en conjunto como cada subsistema, funciona correctamente, usando el sistema distintos usuarios y expertos para comprobar que las recomendaciones sean buenas, evitando errores por comisión (respuestas incorrectas) o por omisión (sin respuesta).

Por todo esto, concluimos que el sistema final es un sistema de calidad.

3. Descripción del sistema

Describiremos el sistema por subsistemas, de manera que analizaremos cada uno por separado donde además, dentro de cada subsistema encontraremos varios módulos que se encargan de distintas tareas.

Siempre habrá un hecho del tipo (MActual ?modulo), donde ?modulo nos indique el modulo que está actualmente cargado, para hacerlo más comprensible los módulos de Recomendar Rama tendrán al final RR, y los módulos de Recomendar Asignaturas tendrán RA.

Además todo lo que no sea parte del conocimiento base que se haya cargado al principio se borra cuando se termina la recomendación, volvemos de menú... esto se hace para evitar información residual que pueda afectar a la hora de ejecutar otra vez las recomendaciones.

Subsistema Menu Principal

Este subsistema esta implementado en el fichero “menu.clp”, con un solo módulo “MMenu”; que sirve como punto de acceso principal para acceder a los dos recomendadores. Se nos muestra un menú simple y se nos pide una opción para elegir un subsistema de recomendación o salir del programa; ante una opción incorrecta vuelve a pedir una opción.

Subsistema Recomendar Rama

Este subsistema está implementado en el fichero “recomendar_rama.clp”, con 3 módulos distintos; que implementa el subsistema que se encarga de extraer información del usuario para recomendarle que mención debería coger. Para la recomendación, primero se extraen las características esenciales del conocimiento establecido en la fase de estudio, que consiste en saber la nota media del estudiante y su interés por temas principales relacionados con cada rama.

Para obtener las recomendaciones se ha usado como base la estructura de árbol obtenido en el estudio previo del conocimiento, de manera que según las distintas respuestas se llega a una conclusión distinta. Están contemplado todos los casos de respuesta, incluso los casos en los que se omiten algunas contestaciones, de manera que siempre se da una manera justificada y acorde a los criterios extraídos.

En concreto, se hacen preguntas al usuario del estilo SI/NO sobre cada tema como “¿te gustan las matemáticas?”, “¿te gusta trabajar con páginas web”, o “dime tu nota media”. Hay una pregunta por cada rama que cubre la temática general y otra por la nota media que discretizamos en 1 (BAJA), 1 (MEDIA), o 2 (ALTA) (los valores de entrada pueden ser continuos o categóricos, que se transforman después), de manera que

en base a las respuestas se aplican como si fuesen las reglas de un árbol (está modificado para contemplar omisión de respuestas). En base al gusto de temáticas nos permite relacionar bien la mención, añadiendo una componente de nota media para dar avisos (no es fundamental la nota si tienes interés, pero el sistema avisa) o para discernir cuando no se tienen los gustos claros.

El orden de procesamiento pasa primero por el módulo de extracción de la información, que utiliza después el módulo de cálculo para obtener una o varias menciones (hay casos que pueden darse varias) que pasa al módulo final de respuesta, que informa al usuario de la respuesta con una justificación.

Nota: se hacen comprobaciones para que los valores que se piden sean correctos, de manera que se repite la pregunta si el usuario realiza una respuesta mal.

Subsistema Recomendar Asignaturas

Este subsistema está implementado en el fichero “recomendar_asignaturas.clp”, con 5 módulos distintos; que implementa el subsistema que se encarga de extraer información del usuario para recomendarle, en base a las asignaturas y el nº de créditos indicado por el usuario, de que asignaturas debería matricularse. Para esta recomendación, se sigue un proceso similar al de la rama: extraemos el conocimiento relevante del usuario, hacemos un procesamiento de la información para obtener una recomendación, y esta es comunicada al usuario.

En este caso, como hay múltiples factores que afectan según la persona y las asignaturas, hemos considerado asignar una puntuación a cada asignatura, de manera que añadimos a la puntuación de cada asignatura en base a los distintos criterios que hemos implementado. En concreto se ha considerado las relaciones:

- Mención del estudiante: si el estudiante tiene mención, es mejor tomar las asignaturas de esta rama, ya que nos asegura plaza y se de temario que guste al estudiante.
- Intereses del estudiante: si los temas generales de la asignatura coinciden con las áreas de mayor interés del estudiante valoramos mejor la asignatura (por ejemplo Software, Hardware, Gráficos...).
- Nota media del estudiante y dificultad de la asignatura: preferimos asignaturas cuyo nivel de dificultad sea razonable para el estudiante, que está relacionado con su nota media.
- Ganas de trabajar del estudiante y carga de trabajo de la asignatura: priorizamos asignaturas cuya cantidad de trabajo no sea superior al interés del estudiante en realizar tareas.
- Prioridad de asignaturas de curso inferior: es preferible hacer asignaturas de curso inferior, ya que suele darse temas básicos sobre los que usar en otras asignaturas superiores; también incluso por comodidad.

- Prioridad de obligatorias frente optativas: las asignaturas optativas tienen menos importancia frente a las obligatorias, ya que estas últimas son bases en la formación del estudiante mientras que las optativas son complementarias.

Para ello consideramos en la base del conocimiento las distintas asignaturas que existen, añadiendo las cualidades subjetivas discretizadas (de forma análoga que en recomendar rama), la carga de trabajo y dificultad, por un experto que valora cada una de ellas, y también extrayendo la información del usuario: su mención, intereses, nota media y ganas de trabajar (además de la lista de asignaturas y el nº de créditos).

Una vez obtenida la información el sistema valora todos estos criterios obteniendo una puntuación para cada asignatura, donde se escogen las asignaturas con mayor puntuación en base a los créditos que se tienen; finalmente informando al usuario de los resultados junto al razonamiento (los factores aplicados).

Además, permitimos que el usuario pueda saltarse todas las preguntas de los criterios, por lo que al encontrarnos con incertidumbre (conocimiento incompleto) tenemos que manejarlo, usando el razonamiento por defecto. Aplicamos los valores por defecto a la información, de manera que decimos que no se tiene mención por defecto, las ganas de trabajar y la nota media son normales; y finalmente los intereses por defecto dependen de la rama que se tenga (según la mención hay unos intereses distintos por defecto) que por ejemplo, los intereses por defecto para no tener mención serían Software, Gráficos e Inteligencia Artificial (áreas más comunes que suelen interesar en general), o Ciencia de Datos, Inteligencia Artificial y Software para la rama CSI.

Los valores por defecto han sido obtenidos del estudio realizado, de manera que no son arbitrarios; además tanto si se sobreescribe un valor por defecto, o se deja el valor por defecto el sistema avisa por pantalla para que el usuario esté informado del razonamiento por defecto que se está produciendo en todo momento. Añadir también que el sistema de razonamiento por defecto se ha adaptado fielmente al visto en clase, de manera que, como ya veremos más adelante, añadiremos **defecto** o **seguro** en los campos de la información cuando sea un valor por defecto o seguro.

Finalmente, el subsistema tiene también un submenú que nos permite listar todas las asignaturas cargadas en la base del conocimiento, volver atrás, o pedir la recomendación.

Nota: se realizan comprobaciones para las respuestas del usuario, para evitar valores incorrectos o no validos; en este caso se para todo si se comete un error.

a. Variables de entrada del problema

Describimos los hechos relacionados con la entrada de información del usuario (recogida de información).

Subsistema Menu Principal

Hecho para recoger la opción del menú:

- (Opcion ?opcion) indica la ?opcion (a, b, c) que se ha escogido del menú.

Subsistema Recomendar Rama

Hechos para recoger las respuestas del usuario a las preguntas del subsistema:

- (Gusta ?tema ?respuesta) indica si al usuario le gusta o no ?respuesta (SI, NO) el tema sobre el que se le ha preguntado ?tema (Hardware, Software, Matematicas...)
- (NotaMedia ?calif) indica la nota media del usuario ?calif discretizada en 1 (BAJA), 2 (MEDIA) y 3 (ALTA).

Subsistema Recomendar Asignaturas

Hecho para recoger la opción del menú:

- (OpcionRA ?opcion) indica la ?opcion (a, b, c, d) que se ha escogido del menú.

Para recoger toda la información del usuario se ha realizado un template llamado Usuario que nos permite encapsular la información recogida:

```
; Clase para encapsular la información introducida por el usuario
(deftemplate Usuario
  ; Lista de asignaturas que necesita recomendación
  (multislot Asignaturas
    (type SYMBOL))

  ; N° de créditos para matricularse
  (slot Credits
    (type INTEGER))

  ; Mención del usuario (si es Comun se interpreta que no tiene)
  ; Por defecto tomamos Comun
  (multislot Mencion
    (type SYMBOL)
    (default Comun defecto))

  ; Nota media del usuario, tomamos 1 (Baja), 2 (Media), 3 (Alta), por defecto es 2
  (multislot NotaMedia
```



```

    (default 2 defecto))

; Ganas de trabajar del usuario, tomamos 1 (Baja), 2 (Media), 3 (Alta),
; por defecto es 2
(multislot CargaTrabajo
  (default 2 defecto))

; Intereses del usuario: Software (S), Hardware (H), Web (W), Bases de Datos (BD),
; Redes (R), Ciencia de Datos (CD), Inteligencia Artificial (IA), Gráficos (G).
; Empresa (E). Por defecto vendrá dado según la mención que se tenga.
(multislot Intereses)
)

```

b. Variables de salida del problema

Recogemos los hechos relacionados con la salida de información del sistema.

Subsistema Recomendar Rama

Hecho de salida para informar de la rama recomendada:

- (Consejo ?rama ?motivo ?experto) nos indica que el sistema ha recomendado la rama ?rama (CSI) dando los motivos ?motivo (“porque te gustan las mates”) aconsejado por el experto ?experto (Miguel).

Subsistema Recomendar Asignaturas

Hecho de salida para informar de asignatura recomendada:

- (AsignaturaRecomendada ?num_asig ?asig \$?exp) nos indica que se recomienda la asignatura ?asig (IC), que ha sido la número ?num_asig (1) en sr escogida, por los motivos indicados en \$?exp (“porque los intereses coinciden...”).

c. Conocimiento global del sistema

Especificamos el conocimiento inicial del sistema que tiene cargado, en general muchas son auxiliares, las de mayor relevancia de conocimiento como tal serían principalmente la información sobre las asignaturas.

Nota: Información detallada sobre los hechos ordenados se puede ver en el apartado 3.e.

Subsistema Menu Principal

Tenemos hechos para listar las opciones y manejarlas facts `HechosMenuInicial`:

```
(deffacts HechosMenuInicial
  (Comando MostrarMenu)
  (MActual MMenu)
  (Opciones a b c)
  (Activar a MHacerPreguntasRR RecomendarRama)
  (Activar b MMenuRA MostrarMenu)
)
```

Subsistema Recomendar Rama

Hechos para preguntar, respuestas válidas y transformación nota media en discreta, en facts `HechosRamaAux`:

```
(deffacts HechosRamaAux
  ; Preguntas
  (PreguntaSINO Mates "Te gustan las mates aplicadas a la informatica (IA, RV...)?")
  (PreguntaSINO Hardware "Te gusta el hardware (robots, procesadores...)?")
  (PreguntaSINO BigData "Te gusta el bigdata (ciencia de datos, analisis...)?")
  (PreguntaSINO Software "Te gusta programar software (juegos, programas...)?")
  (PreguntaSINO Web "Te gusta trabajar con paginas web? (código, estilo, bbdd...)?")
  ; Resuestas válidas
  (RespuestasSINO SI NO NS STOP)
  ; Nota media categorica a discreta
  (VarCategoricaRR BAJA 1)
  (VarCategoricaRR MEDIA 2)
  (VarCategoricaRR ALTA 3)
  (VarCategoricaRR STOP STOP)
  (VarCategoricaRR NS NS)
  ; Nota media continua a discreta
  (VarNumericaRR 0 6 1)
  (VarNumericaRR 6 7.5 2)
  (VarNumericaRR 7.5 10.1 3)
)
```

Subsistema Recomendar Asignaturas

Para almacenar la información de las asignaturas usamos un `template` llamado `Asignatura`:

```

; Clase que representa una Asignatura
(deftemplate Asignatura
  ; ID de la asignatura: generalmente tomamos la abreviatura
  (slot ID
    (type SYMBOL))
  ; Nombre de la asignatura
  (slot Nombre
    (type STRING))
  ; Curso de la asignatura: de 1 a 4
  (slot Curso
    (type INTEGER)
    (range 1 4))
  ; Cuatrimestre de la asignatura: 1 o 2
  (slot Cuatrimestre
    (type INTEGER)
    (range 1 2))
  ; N° de creditos de la asignatura: la mayoría dan 6, así que se pone por defecto
  ; Además restringimos al n° máximo de créditos posible (18) y mínimo (1)
  (slot Creditos
    (type INTEGER)
    (range 1 18)
    (default 6))
  ; La rama de la asignatura: si es para todos se toma Comun
  ; Por defecto se toma Comun
  (slot Rama
    (type SYMBOL)
    (allowed-values CSI IC IS SI TI Comun)
    (default Comun))
  ; Dificultad de la asignatura: 1 (facil), 2 (media), 3 (dificil), por defecto 2
  (slot Dificultad
    (type INTEGER)
    (range 1 3)
    (default 2))
  ; Carga de trabajo de la asignatura: 1 (poco), 2 (medio), 3 (mucho), por defecto 2
  (slot Carga
    (type INTEGER)
    (range 1 3)
    (default 2))
  ; Tipo de asignatura: obligatoria u optativa, por defecto es obligatoria
  (slot Tipo
    (type SYMBOL)
    (allowed-values Obligatoria Optativa)
    (default Obligatoria))
  ; Temas principales dados en la asignatura: Software (S), Hardware (H), Web (W),

```

```

; Bases de Datos (BD), Redes (R), Ciencia de Datos (CD), Inteligencia Artificial (IA),
; Gráficos (G), Empresa (E)
(multislot Temas
  (type SYMBOL)
  (allowed-values S H W BD R CD IA G E)))

```

Incluimos una lista extensa de asignaturas en `facts ListaAsignaturas`, variada para tener unas cuantas de todos cursos y cuatrimestres, sobre todo en 3º y 4º, por ejemplo dos hechos de la lista:

```

(Asignatura (ID FP)
  (Nombre "Fundamentos de la Programacion")
  (Curso 1)
  (Cuatrimestre 1)
  (Temas S)
  (Dificultad 2)
  (Carga 2))

(Asignatura (ID FFT)
  (Nombre "Fundamentos Fisicos y Tecnologicos")
  (Curso 1)
  (Cuatrimestre 1)
  (Temas H)
  (Dificultad 3)
  (Carga 3))

```

Hechos para manejar el submenú y sus opciones (`facts HechosMenu`):

```

(deffacts HechosMenu
  (OpcionesRA a b c d)
  (ActivarRA a MListarRA ListarAsignaturas)
  (ActivarRA b MPreguntarRA HacerPreguntasAsignaturas)
  (ActivarRA c MMenu MostrarMenu)
)

```

Templates Rama y Tipo auxiliares para imprimir la lista de asignaturas, junto a los hechos auxiliares para cada tipo:

```

; Clase auxiliar para listar
(deftemplate Rama
  (slot ID
    (type SYMBOL)

```

```

        (allowed-values CSI IC IS SI TI Comun))
)

; Clase auxiliar para listar
(deftemplate Tipo
  (slot ID
    (type SYMBOL)
    (allowed-values Obligatoria Optativa))
)

; Hechos auxiliares para listar
(deffacts HechosAuxiliaresListar
  ; Ramas
  (Rama (ID Comun))
  (Rama (ID CSI))
  (Rama (ID IC))
  (Rama (ID IS))
  (Rama (ID SI))
  (Rama (ID TI))
  ; Tipo asignatura
  (Tipo (ID Obligatoria))
  (Tipo (ID Optativa))
)

```

Finalmente hechos auxiliares para comprobar que las respuestas del usuario sean correctas, y hacer las transformaciones de los valores numéricos continuos/categoricos en numéricos discretos (facts HechosAuxiliaresPreguntar):

```

(deffacts HechosAuxiliaresPreguntar
  ; Intereses temáticos
  (Interes S H W BD R CD IA G E)
  ; Intereses por defecto según mención
  (InteresesDefecto Comun S G IA)
  (InteresesDefecto CSI CD IA S)
  (InteresesDefecto IS S G IA)
  (InteresesDefecto IC H R)
  (InteresesDefecto SI CD BD W)
  (InteresesDefecto TI W R BD)
  ; Nota media / carga categorica a discreta
  (VarCategorica BAJA 1)
  (VarCategorica MEDIA 2)
  (VarCategorica ALTA 3)
  ; Nota media continua a discreta

```

```

(VarNumerica NotaMedia 0 6 1)
(VarNumerica NotaMedia 6 7.5 2)
(VarNumerica NotaMedia 7.5 10.1 3)
; Ganas trabajar a discreta
(VarNumerica CargaTrabajo 0 2.5 1)
(VarNumerica CargaTrabajo 2.5 3.5 2)
(VarNumerica CargaTrabajo 3.5 5.1 3)
)

```

d. Especificación de los módulos

i. Estructura de módulos

Como ya he comentado, se ha realizado distintos módulos dentro de cada subsistema, y se controla el módulo activo con el hecho (`MActual ?modulo`). La estructura es la siguiente:

- El subsistema **Menu Principal** está compuesto unicamente por un único módulo `MMenu`, implementado en el archivo `menu.clp`.
- El subsistema **Recomendar Rama** está compuesto por 3 módulos: `MHacerPreguntasRR`, `MCalcularRamaRR` y `MRespuestaRamaRR`, que se ejecutan secuencialmente; implementado en el archivo `recomendar_rama.clp`
- El susistema **Recomendar Asignaturas** está compuesto por 5 módulos: `MMenuRA`, `MListarRA`, `MPreguntarRA`, `MCalcularRA` y `MResultadosRA` (estos 3 últimos se ejecutan secuencialmente); implementado en el archivo `recomendar_asignaturas.clp`.

ii. Descripción del módulo

Subsistema Menú Principal El módulo `MMenu` es un módulo auxiliar que se encarga de mostrar al usuario el menú con las opciones, cuyo objetivo es procesar correctamente la opción seleccionada (salir o cambiar de subsistema), manejando como conocimiento las opciones disponibles y los comandos a realizar; no deduce nada nuevo.

Subsistema Recomendar Rama El módulo `MHacerPreguntasRR` se encarga de preguntar al usuario las preguntas para extraer la información, comprobar las respuesta, y cuando el usuario para o se acaban las preguntas pasar al módulo `MCalcularRamaRR`. Su objetivo por tanto es obtener la información del usuario, usando como conocimiento las preguntas que le hace al usuario, sin deducir nada nuevo.

El módulo **MCalcularRamaRR** es el módulo que utiliza la información extraída del módulo anterior para obtener una recomendación. Por tanto el objetivo es realizar una recomendación, usando como conocimiento la información extraída del usuario deduciendo nuevo conocimiento: la rama recomendada.

Finalmente el módulo **MRespuestaRamaRR** informa al usuario de la rama recomendada dando los motivos de por que se ha hecho esto, usando el conocimiento deducido del módulo anterior aunque este no deduce nada nuevo.

Subsistema Recomendar Asignaturas El módulo **MMenuRA** tiene una funcionalidad igual que **MMenu**: muestra al usuario el menú con las opciones procesando el cambio de módulo, manejando las opciones disponibles sin deducir nada nuevo.

El módulo **MListarRA** se encarga de imprimir por pantalla todas las asignaturas que se tienen en la base del conocimiento, por curso, cuatrimestre, rama, tipo... para que el usuario sepa que asignaturas puede pedir recomendación; no deduce nada nuevo.

El módulo **MPreguntarRA** funciona igual que **MHacerPreguntasRR**: su función es la de extraer el conocimiento del usuario mediante preguntas. No hay orden fijo para la realización (la única es que la mención se haya tomado antes que los intereses) y también se encarga de chequear que la información obtenida sea correcta.

El módulo **MCalcularRA** también es como **MCalcularRamaRA**: utiliza el conocimiento extraído anteriormente para deducir las asignaturas recomendadas, mediante el cálculo de puntuaciones en base a los criterios ya mencionados.

Por último, el módulo **MResultadosRA** que es igual que **MResultadosRA**, informa al usuario de las asignaturas recomendadas (conocimiento deducido del módulo anterior), mostrando las asignaturas por orden según la puntuación y con las razones.

e. Hechos y reglas

Lista de hechos y reglas en cada módulo con breve explicación, las reglas se ponen sin paréntesis y los hechos sí, usando siempre (**MActual ?modulo**) el hecho que controla el módulo que está activo:

Subsubsistema Menú Principal

MMenu

- (**Comando ?comando**) es un hecho multipropósito para indicar que se active/haga algo.
- (**Comando MostrarMenu**) indica cuando se debe imprimir el menú.
- **MostrarMenuPrincipal** imprime el menú principal y recoge la respuesta del usuario en (**Opcion ?opt**).

- (Opcion ?opt): opción del usuario elegido.
- ChequearOpcionPrincipal comprueba que (Opcion ?opt) sea válida si ?opt está en (Opciones \$?opts), si no se imprime de nuevo el menú.
- (Opciones \$?opts): distintas opciones disponibles del menú.
- EjecutarOpcionPrincipal si (Opcion ?opt) es correcto ejecuta (Activar ?opt ?modulo ?comando).
- (Activar ?opt ?modulo ?comando): indica que al usar la opción ?opt se cambie al módulo ?modulo añadiendo el hecho (Comando ?comando).
- SalirPrincipal sale del programa si la opción es c.

Subsistema Recomendar Rama

MHacerPreguntasRR

- (Comando RecomendarRama) se encarga de activar la explicación inicial y crear las preguntas.
- HacerExplicacionInicial muestra el mensaje inicial y la explicación para el usuario, añadiendo los hechos (HacerPregunta ?tema).
- (HacerPregunta ?tema) nos indica que se debe hacer una pregunta del tema ?tema.
- (PreguntaSINO ?tema ?pregunta) es la información de la pregunta con el ?tema y la pregunta ?pregunta.
- Preguntar_SI_NO realiza una pregunta de SI o NO, recogiendo el resultado en (Gusta ?tema ?res).
- (Gusta ?tema ?res) indica la respuesta ?res para el tema ?tema.
- Chequear_SI_NO vuelve a preguntar si la respuesta no ha sido válida, es decir si no está en (RespuestasSINO \$?res).
- (RespuestasSINO \$?res) son las respuestas válidas para SI/NO \$?res.
- Acabar_SI_NO acaba las preguntas si se responde con STOP.
- Preguntar_NotaMedia pregunta al usuario por su nota media guardada en (NotaMedia ?nota check).
- (NotaMedia ?nota check) indica la nota media ?nota y si tiene check es que aun tiene que ser transformada.
- Chequear_NotaMedia comprueba que la nota media numérica esté en $[0, 10]$, si no pregunta de nuevo.
- Chequear_NotaCategorica comprueba que sea uno de los símbolos aceptados con (VarCategoricaRR ?nota ?), si no pregunta de nuevo.
- (VarCategoricaRR ?nota ?nueva_nota) indica que el símbolo ?nota se transforma en ?nueva_nota.
- Convertir_NotaCategorica convierte la nota categórica correcta a discreta numérica con (VarCategoricaRR ?nota ?), dejando (NotaMedia ?nueva_nota).
- Convertir_NotaContinua convierte la nota continua a discreta usando (VarNumericaRR ?lim_inf ?lim_sup ?nueva_nota), dejando (NotaMedia

?nueva_nota).

- (VarNumericaRR ?lim_inf ?lim_sup ?nueva_nota) indica que si el número continuo está en (?lim_inf, ?lim_sup] se transforma en número discreto ?nueva_nota.
- Acabar_NotaMedia si la respuesta ha sido STOP, se pasa directamete al cálculo de rama.
- SinPreguntas pasa a calcular ramas si ya no quedan mas (HacerPregunta ?).

MCalcularRamaRR

- (Consejo ?rama ?razon ?experto) representa el consejo de coer la rama ?rama por los motivos ?razon del experto ?experto.
- (Aviso ?rama) crea un aviso, de mirar las notas, para la rama ?rama.
- GustaHardware recomienda IC para quien le guste Hardware, crea aviso.
- GustaMatematicas recomienda CSI para quien le guste Mates pero no Hardware, crea aviso.
- GustaBigdata recomienda SI si te gusta Bigdata pero no Mates/Hardware.
- GustaSoftware recomienda IS si te gusta Software pero no Mates/Hardware/Bigdata, crea aviso.
- GustaWeb recomienda TI si le gusta Web y no otra cosa.
- NoGustaNada recomienda IS si no le gusta nada.
- SoloNotas_Baja recomienda TI/SI si solo tiene notas bajas.
- SoloNotas_Media recomienda IS/SI/IC si solo tiene notas medias.
- SoloNotas_Alta recomienda CSI si se tienen notas altas.
- SinRespuestas recomienda IS si no se ha respondido a nada.
- Aviso añade al consejo de rama, una advertencia si la nota es BAJA y había un (Aviso ?rama).
- RamaCalculada cambia al siguiente módulo y añade el hecho (Experto ?experto).
- (Experto ?experto) el nombre del experto, para imprimirlo aparte del consejo.

MRespuestaRamaRR

- RespuestaExperto mensaje de respuesta para el usuario aconsejado por el experto
- ImprimeConsejo imprime los consejos
- LimpiarHechosRama limpia (Experto ?) y añade (Comando LimpiarRama) para limpiar los hechos.
- LimpiarConsejos limpia los hechos (Consejo \$?)
- LimpiaGustos limpia los hechos (Gusta \$?)
- LimpiaPreguntas limpia los hechos (HacerPregunta \$?)
- LimpiaNotaMedia limpia el hecho (NotaMedia \$?)
- LimpiaAvisos limpia los hechos (Aviso \$?)
- VolverMenuPrincipal cambia de modulo al menú principal mostrandolo.

Subsistema Recomendar Asignaturas

MMenuRA

- (Comando MostrarMenu) indica que hay que mostrar el menú.
- MostrarMenu muestra el menú de recomendar asignaturas, y recoge la opción del usuario en (OpcionRA ?opt).
- (OpcionRA ?opt) opción escogida por el usuario.
- ChequearOpcion imprime de nuevo el menú si la opción no está en (OpcionesRA \$?opts).
- (OpcionesRA \$?opts) las opciones válidas del menú.
- EjecutarOpcion ejecuta la opción con (ActivarRA ?opt ?modulo ?comando) cambiando al módulo ?modulo añadiendo el hecho (Comando ?comando).
- Salir si la opción es la d, se sale del programa.
- (Comando VolverMenu) indica que se tiene que volver al menú principal.
- VolverMenu vuelve al menú principal cambiando de módulo.

MListarRA

- (Asignatura ID Nombre Curso Cuatrimestre Rama Dificultad Carga Tipo Temas) es un template que representa a una Asignatura, guardando todos los valores necesarios para aconsejar asignaturas.
- (Rama ID) es un template auxiliar para listar las asignaturas.
- (Tipo ID) otro template auxiliar para lo mismo.
- ListarAsignaturas lista las asignaturas por curso, cuatrimestre, tipo y rama, añadiendo pausas con ENTER para que sea legible; automáticamente vuelve al menú.

MPreguntarRA

- (Usuario Asignaturas Creditos Mencion NotaMedia CargaTrabajo Intereses) es un template para guardar la información que se obtiene del usuario.
- (Comando HacerPreguntasAsignaturas) indica que se empieza a recomendar asignaturas.
- RecomendadorAsignaturas muestra mensaje informativo y añade las preguntas que se van a hacer (Preguntar ?) y añade un hecho vacío del usuario.
- (Preguntar ?tipo) indica que se debe hacer una pregunta de tipo ?tipo.
- SalidaRecomendadorPreguntas se encarga de limpiar (Preguntar ?) cuando se vuelve al menú.
- SalidaRecomendadorUsuario lo mismo pero limpia (Usuario).
- PreguntaListaAsignaturas pregunta por la lista de asignaturas del usuario, y las guarda en (Asignaturas ?asigs); si se indica STOP se vuelve al menú.
- (Asignaturas ?asigs) hecho para chequear que la lista es correcta.
- ParadaListaAsignaturas si hay un STOP, inicia (Comando VolverMenu).

- `ChequearAsignaturas` comprueba que todas las asignaturas de (`Asignaturas ?asigs`) estén, si no da error y vuelve al menú.
- `EliminarAsignaturasRepetidas` elimina asignaturas repetidas de (`Asignaturas ?asigs`).
- `AsignaturasVacias` da error si la lista de asignaturas está vacía.
- `AsignaturasCorrecto` añade a (`Usuario`) la lista de asignaturas si no hay fallos.
- `PreguntarCreditos` pide el nº de créditos a matricular al usuario y lo guarda en (`Creditos ?cred`).
- (`Creditos ?cred`) nº de créditos a chequear.
- `ComprobarCreditosEntero` comprueba que se ha pasado un nº entero de créditos.
- `ComprobarCreditosValido` comprueba que es un nº positivo de créditos, si lo es añade a (`Usuario`).
- `PreguntarMencion` pregunta por la mención del usuario y lo guarda en (`Mencion ?mencion`), se puede saltar.
- (`Mencion ?mencion`) mención respuesta por comprobar.
- `MencionVacía` no se ha dado mención, se avisa de la mención por defecto (`Usuario (Mencion Comun defecto)`) y añade (`Preguntar Intereses`).
- `ComprobarMencion` comprueba que la mención existe.
- `MencionCorrecta` añade a (`Usuario (Mencion ?mencion seguro)`) la mención, y avisa de que se retracta el razonamiento por defecto, añade (`Preguntar Intereses`).
- `PreguntarNotaMedia` pregunta la nota media al usuario guardandolo en (`NotaMedia ?nota`), se puede saltar.
- (`NotaMedia ?nota`) la nota media del usuario sin verificar.
- `NotaVacía` si se ha pasado la pregunta, se toma por defecto (`Usuario (NotaMedia 2 defecto)`) y se avisa.
- `ComprobarNotaNumerica` si la nota es numerica se comprueba que esté en $[0, 10]$, si no da error y vuelve al menú.
- `ComprobarNotaCategorica` lo mismo pero con categórica, usando (`VarCategorica ?nota ?`).
- (`VarCategorica ?nota ?nota_nueva`) indica que para el valor categórico `?nota` se debe tomar el valor numérico discreto `?nota_nueva`.
- `TransformarNotaCategorica` transforma la nota categórica a discreta con (`VarCategorica ?nota ?nueva_nota`) y añade al usuario, avisando que se retracta la lógica por defecto (`NotaMedia ?nueva_nota seguro`).
- `TransformarNotaDecimal` lo mismo pero con nota continua usando (`VarNumerica NotaMedia ?lim_inf ?lim_sup ?nueva_nota`).
- (`VarNumerica ?var ?lim_inf ?lim_sup ?nueva_nota`) indica que para la variable `?var`, un número en continua que esté en el intervalo $[\text{?lim_inf}, \text{?lim_sup}]$ se transforma en la variable numérica discreta `?nueva_nota`.
- `PreguntarCargaTrabajo` igual que `PreguntarNotaNumerica` pero con la carga de trabajo que quiere el usuario, se puede saltar.
- `CargaVacía` igual que `NotaVacía`.
- `ComprobarCargaNumerica` igual que `ComprobarNotaNumerica`.

- `ComprobarCargaCategorica` igual que `ComprobarNotaCategorica`.
- `TransformarCargaCategorica` igual que `TransformarNotaCategorica`.
- `TransformarCargaDecimal` igual que `TransformarNotaDecimal`.
- `PreguntarIntereses` pregunta al usuario por los intereses que tiene y se guardan en `(Intereses $?intereses)`, se puede saltar.
- `ChequearIntereses` comprueba que la lista de intereses es válida con `(Interes $?intereses)`, si no da error y vuelve al menú.
- `(Intereses $?intereses)` intereses del usuario para comprobar.
- `(Interes $?intereses)` lista de intereses disponibles.
- `EliminarInteresesRepetidos` elimina los intereses repetidos.
- `InteresesVacios` si se ha saltado la pregunta se añade `(Comando InferirIntereses)` para aplicar razonamiento por defecto.
- `InferirIntereses` pone los intereses por defecto según la mención del usuario, de manera que usando `(InteresesDefecto ?mencion $?intereses)` se modifica `(Usuario (Intereses $?intereses defecto))` y se avisa del razonamiento por defecto.
- `InteresesCorrectos` si todos los intereses indicados por el usuario son correctos, se retracta la lógica por defecto y se modifica `(Usuario (Intereses $?intereses seguro))`.
- `FinInformacion` cuando todas las preguntas se acaben pasamos al modulo `MCalcularRA` añadiendo los hechos `(CreditosActuales ?cred)`, `(HayOptativas NO)`, `(ASignatura 1)`, `(CursoMaxMin 0 0)`.

MCalcularRA

- `(HayOptativas ?res)` nos dice si hay optativas o no de las asignaturas del usuario.
- `(CursoMaxMin ?max ?min)` nos dice el curso máximo y mínimo de las asignaturas del usuario.
- `(ASignatura ?n)` nos dice que hay `?n - 1` asignaturas escogidas.
- `(CreditosActuales ?cred)` nos dice que faltan por matricular `?cred` créditos.
- `IniciarPuntuaciones` inicia las puntuaciones de todas las asignaturas con `(Puntuacion ?asig ?punt $?explicacion)`, y añade `(Puntuar ?tema ?asig)` para realizar el cálculo de puntuaciones.
- `(Puntuacion ?asig ?punt $?explicacion)` indica que la asignatura `?asig` tiene una puntuación de `?punt` por los motivos `$?explicacion`.
- `(Puntuar ?tema ?asig)` indica que hay que calcular la puntuación de `?tema` a la asignatura `?asig`.
- `PuntuacionMencion` añade 1 a las asignaturas cuya rama coincide con la mención de `(Usuario)` y añade explicación; solo aplica si se tiene mención.
- `PuntuacionDificultad` añade 1 a las asignaturas cuya dificultad sea menor o igual que la nota media de `(Usuario)` añadiendo explicación.
- `PuntuacionCarga` añade 1 a las asignaturas cuya carga de trabajo sea menor o igual que las ganas de trabajar de `(Usuario)` añadiendo explicación.

- `ComprobarOptativas` se encarga de actualizar (`HayOptativas ?res`) si hay optativas.
- `PuntuacionTipo` añade 1 a las asignaturas obligatorias añadiendo explicación, solo en el caso de que haya al menos una optativa.
- `ComprobarCursoMin` actualiza el curso mínimo en (`CursoMaxMin ?max ?min`).
- `ComprobarCursoMax` actualiza el curso máximo en (`CursoMaxMin ?max ?min`).
- `PuntuacionCurso` añade 2 a las asignaturas cuyo curso sea el menor de las asignaturas, añadiendo explicación, solo si hay al menos dos asignaturas en dos años distintos.
- `PuntuacionIntereses` crea un hecho con los temas de la asignatura (`ComprobarIntereses ?asig $?temas`) para comprobar las coincidencias de los del (`Usuario`).
- `InteresesEsta` si el primer tema de `ComprobarIntereses ? $?temas` está en la lista de intereses del usuario se añade 2 a la puntuación, además se quita el tema de la lista.
- `InteresesNoEsta` si el primer tema no esta, se quita de la lista.
- `FinInteresesLista` cuando la lista se acaba borramos el hecho.
- `CalcularAsignaturasRecomendadas` cuando ya están todas las puntuaciones, mientras queden créditos para matricular con (`CreditosActuales ?cred`) se manda (`Comando AsignaturaMax`) para escoger una asignatura.
- `ObtenerAsignaturaMax` obtiene la asignatura con mayor puntuación (cuya resta al nº de créditos por matricular sea posible), actualiza (`CreditosActuales ?cred`) restandole los créditos de la asignatura seleccionada, añade (`AsignaturaRecomendada ?num ?asig $?exp`) y modifica (`NAsignatura ?num`) añadiendo 1 más.
- (`AsignaturaRecomendada ?num ?asig $?exp`) indica que la asignatura `?asig` es la asignatura nº `?num` en ser elegida por las razones `$?exp`.
- `FinCreditos` cuando ya no se pueden coger más asignaturas (no cuadran el nº de créditos, o es 0) se lanza (`Comando LimpiarHechos`) para limpiar hechos; además avisa si han quedado créditos por asignar.
- `LimpiarHechosPuntuacion` limpia los hechos (`Puntuacion $?`)
- `LimpiarHechosExtra` limpia los hechos (`HayOptativas $?`) y (`CursoMaxMin $?`)
- `LimpiarHechosFin` cuando se ha limpiado todo, se pasa al siguiente módulo y se añaden los hechos (`Comando MostrarAsignaturasInicio`) y (`MostrarAsignatura 1`).

Módulo MResultadosRA

- (`Comando MostrarAsignaturasInicio`) indica que se inicie el mostrar las recomendaciones.
- (`MostrarAsignatura ?num`) indica el nº de la asignatura que tiene que mostrarse.
- `MostrarAsignaturasInicial` mensaje para el usuario de que van a imprimirse las asignaturas,
- (`Comando MostrarAsignaturas`) comando para mostrar las recomendaciones.

- **MostrarAsignaturaSinMotivos** aviso de que la asignatura ha sido recomendada por descarte (sin motivos).
- **MostrarAsignatura** muestra la recomendación de la asignatura, añade (**Razones \$?explicaciones**) para añadir las explicaciones.
- (**Razones \$?explicaciones**) las explicaciones de coger esa asignatura, que se deben imprimir.
- **ImprimirRazones** imprime la primera explicación de la lista, y la quita de la lista.
- **RazonesFin** cuando ya no hay más explicaciones, aumentamos (**MostrarAsignatura ?num**) en 1 para imprimir la siguiente asignatura.
- **FinRecomendar** cuando ya no hay más asignaturas que mostrar mandamos (**Comando VolverMenu**) y limpiamos los hechos (**Comando MostrarAsignatura**), (**MostrarAsignatura \$?**), (**NAsignatura ?**) y la información del usuario (**Usuario**).
- **LimpiarRecomendaciones** limpia los hechos (**AsignaturaRecomendada \$?**) antes de volver al menú.

4. Breve manual de uso del sistema

Para poder usar el sistema experto, primero abrimos CLIPS y cargamos los tres archivos con (load 'menu.clp'), (load 'recomendar_asignaturas.clp') y (load 'recomendar_rama.clp'); después hacemos (reset) y finalmente (run).

Se nos mostrará el menú inicial, donde podemos salirnos del programa poniendo **c**, usar el recomendador de rama con **a** o de asignaturas con **b**.

Si ponemos **a**, el recomendador de rama se inicia y nos da un mensaje de información. Después empiezan las preguntas que podemos responder con lo que se nos diga: (numero, **Baja**|**Media**|**Alta**) para NotaMedia y en el resto de preguntas **si** o **no**. Además podemos responder **NS** si no queremos responder o no sabemos, o **STOP** para que nos recomiende directamente (si ponemos algo mal se nos vuelve a pedir). Cuando no haya mas preguntas o pongamos **STOP** el sistema nos mostrará que es lo que nos recomienda, y volveremos al menú principal.

Ahora ponemos **b** y pasamos al submenú del recomendador de asignaturas, que nos permite listar las asignaturas con **a**, iniciar el recomendador con **b**, volver al menú anterior **c** o salir **d**.

Si ponemos **a**, y pulsando ENTER iremos pasando los distintos cursos/cuatrimestres con las asignaturas cargadas (siglas y su nombre completo).

Finalmente si ponemos **b** empieza el recomendador de asignaturas, que nos muestra el mensaje inicial y empieza con las preguntas. Excepto la lista de asignaturas y créditos, todas las preguntas se pueden saltar con ENTER, de manera que nos informa del razonamiento por defecto; además si metemos una respuesta mal se nos indica del error y por qué. Cuando acabe de responder se nos informa de las asignaturas recomendadas, con la lista de razones y volvemos al submenú.

Finalmente en todos los subsistemas limpian los hechos deducidos y auxiliares para que se pueda volver a usar los recomendadores sin problema alguno.