

Doble Grado en  
Ingeniería Informática y Matemáticas

INGENIERÍA DEL CONOCIMIENTO  
(E. Computación y Sistemas Inteligentes)

**PRÁCTICA FINAL: Asesorar a un alumno de Ingeniería  
Informática para elegir Rama y Asignaturas**



**UNIVERSIDAD  
DE GRANADA**

Carlos Santiago Sánchez Muñoz

Grupo de prácticas 1 - Viernes

*DNI:* 75931715K

*Email:* carlossamu7@correo.ugr.es

*28 de mayo de 2020*

# Índice

<b>1. Resumen de cómo funciona el sistema experto</b>	<b>2</b>
<b>2. Descripción del proceso seguido</b>	<b>3</b>
2.1. Procedimiento seguido para el desarrollo de la base de conocimiento . . . . .	3
2.2. Procedimiento de validación y verificación del sistema . . . . .	4
<b>3. Descripción del sistema</b>	<b>4</b>
3.1. Variables de entrada del problema . . . . .	4
3.2. Variables de salida del problema . . . . .	5
3.3. Conocimiento global del sistema . . . . .	6
3.4. Especificación de los módulos desarrollados . . . . .	7
3.5. Hechos y las reglas de cada módulo . . . . .	8
<b>4. Breve manual de uso del sistema</b>	<b>10</b>

## 1. Resumen de cómo funciona el sistema experto

Se ha desarrollado un Sistema Experto para asesorar a alumnos del Grado en Ingeniería Informática respecto a dos temas: la elección de **Ramas** y la elección de **Asignaturas** a matricular.

El Sistema Experto requiere razonamiento humano. De esta manera el primer paso es adquirir conocimiento para la toma de decisiones apoyado en técnicas de razonamiento. En concreto, se ha usado lógica por defecto. La implementación está realizada en CLIPS.

El sistema da la bienvenida al usuario y le pide que indique si quiere asesoramiento acerca de las **Ramas** o de las **Asignaturas**. El sistema está adecuadamente modularizado y en ese momento se centrará en uno de los dos ámbitos. A partir de ese momento realiza unas preguntas diferentes en función de dicha elección con la intención de conocer las preferencias, gustos y habilidades del usuario.

Para el caso de **Ramas** las preguntas son:

- ¿Te gustan las matemáticas? (Si/No)
- ¿Eres trabajador? (Mucho/Normal/Poco)
- ¿Te gusta programar? (Si/No/No se)
- ¿Te gusta el hardware? (Si/No/No se)
- ¿En que te gustaría trabajar? (Pública/Privada/Docencia/No se)
- Introduce tu calificación media de expediente.

Para el caso de **Asignaturas** se ha hecho distinción entre los cuatrimestres para poder dar un consejo más elaborado. Las preguntas son:

- Indique, usando los códigos, las asignaturas que quiere que se consideren (sepárelas por espacios).
- Indique el numero de asignaturas a matricular del 1 cuatrimestre.
- Indique el numero de asignaturas a matricular del 2 cuatrimestre.
- ¿Prefiere asignaturas con mas teóricas o mas prácticas? (Teórica/Práctica)
- ¿Prefiere realizar asignaturas fáciles o difíciles? (Fácil/Difícil)
- ¿Prefiere realizar asignaturas con mucho o poco trabajo? (Mucho/Poco)

En la todas las preguntas, a excepción de las primeras para situar el sistema, se permite responder **Fin** para acabar la ejecución y obtener el consejo final. Asimismo en las preguntas que se indica se permiten respuestas parciales del tipo **No se**.

Por último lugar, el sistema es consistente y robusto ya que almacena el asesoramiento que el usuario ha recibido y cuando finaliza pregunta al usuario si quiere consejo acerca de la otra opción (la que no eligió). Una vez que se ha recibido información tanto de **Ramas** como de **Asignaturas** la ejecución finaliza de manera inevitable.

## 2. Descripción del proceso seguido

En este apartado se va a analizar el procedimiento seguido para el desarrollo de la base del conocimiento y posteriormente el de validación y verificación del sistema.

### 2.1. Procedimiento seguido para el desarrollo de la base de conocimiento

En el desarrollo de la base del conocimiento se ha procurado que la representación del conocimiento sea intuitiva y que los hechos expliquen con claridad y robustez el conocimiento que representan. El conocimiento del sistema representa el estado del asesoramiento en cada momento.

En la adquisición del conocimiento se ha llevado a cabo una educación del mismo mediante una entrevista realizada en la parte de teoría con un experto en el conocimiento, un compañero. Para la parte de **Ramas** se ha usado el árbol realizado en otra de las actividades y en último lugar para el conocimiento de las **Asignaturas** se han usado algunas de las características bipolares del ejercicio de la Rejilla. En particular se han usado el carácter de la asignatura (**Teórica/Práctica**), la dificultad de la asignatura (**Difícil/Fácil**) y la cantidad de trabajo (**Mucho/Poco**).

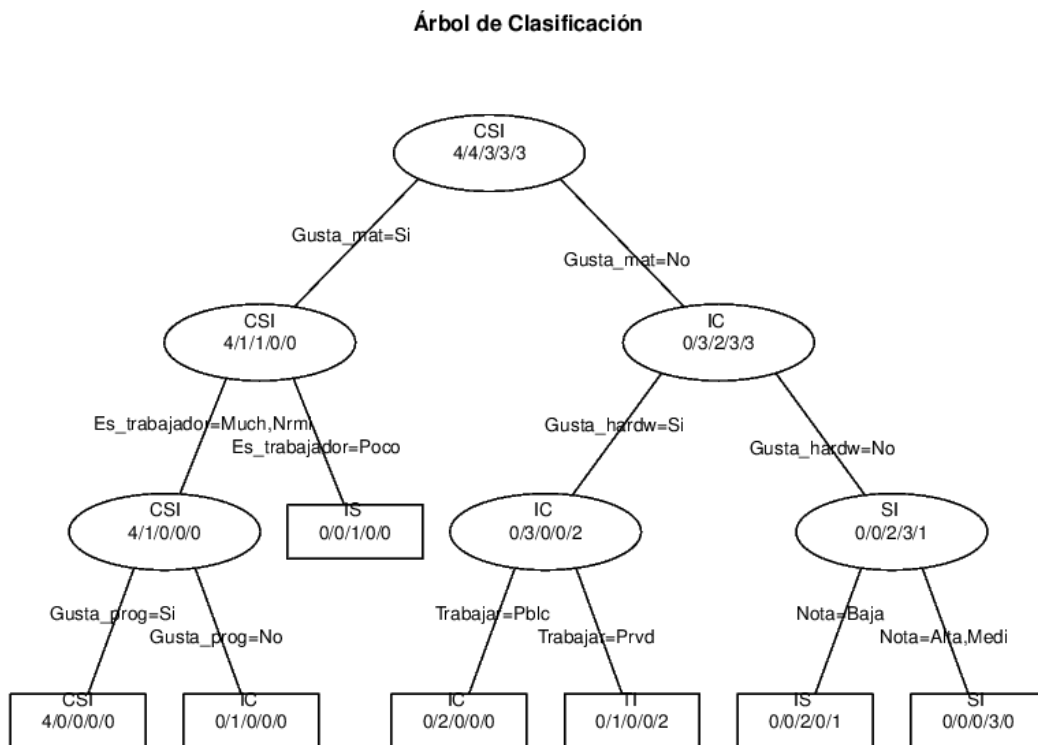


Imagen 1: Árbol de decisión

De esta manera el estudio de propiedades de las **Ramas** y **Asignaturas** permite establecer las cuestiones a formular al usuario para ver si coincide con unas disyunciones u otras en función de eso establecer con cada pregunta un mejor consejo y un razonamiento más elaborado.

## 2.2. Procedimiento de validación y verificación del sistema

Es fundamental realizar este paso con cautela ya que si en el proceso de verificación y validación se cometen errores, esto nos trae como consecuencia un sistema basado en conocimiento mal estructurado, y por lo tanto un sistema experto con errores, incompleto.

Para **validar** el sistema se han realizado testeos exhaustivos en los dos módulos para asegurar que las especificaciones de requisitos se cumplen. El sistema es autoexplicativo e indica al usuario todas las posibilidades en cada momento a la vez que filtra sus respuestas para solo introducir aquellas que se permitan. De este modo la comunicación usuario-sistema es adecuada satisfaciendo las necesidades planteadas y dando consejos y argumentos al usuario.

En resumidas cuentas la representación elegida es adecuada y el sistema es fácil de mantener y de comprender por un usuario general y más aún por un estudiante de Ingeniería Informática. Las comprobaciones de conformidad del producto con las necesidades, los requisitos, y/o las especificaciones definidas por el usuario son positivas convirtiendo en válido el sistema. Lamentablemente, debido a las circunstancias la validación la han realizado personas de mi entorno familiar.

Para **verificar** el sistema vamos a comprobar que sea completo, preciso y consistente. Es consistente porque no se llega a conclusiones incoherentes, es decir, en ningún momento se aserta o retracta un hecho del sistema produciendo contradicciones. Es preciso porque hay corrección en la sintaxis y no hay errores morfológicos. También es completo pues no hay lagunas en la capacidad deductiva (casos donde el sistema no respondería) ya que el número de casos es lo suficientemente bajo como para poder comprobar la totalidad de los estados del sistema.

## 3. Descripción del sistema

Esta sección está dedicada a la presentación del sistema y cómo se ha implementado. Para ello se van a examinar y explicar las variables de entrada y salida, conocimiento global, una especificación de los módulos y los hechos y reglas de cada módulo.

### 3.1. Variables de entrada del problema

La primera variable de entrada es en el módulo principal, el módulo **MAIN**. El usuario indica si quiere **Ramas** o **Asignaturas** y el flujo se redigirá a los módulos **RAMAS** y **ASIGNATURAS** respectivamente. Para ello se usa (`focus MODULO`) y además se aserta en el sistema el hecho (`Elige ?op`) donde las opciones son 0 (por defecto, en el **MAIN**), 1 (ha elegido **RAMAS**) y 2 (ha elegido **ASIGNATURAS**).

Para el caso de **Ramas** las variables de entrada a considerar están relacionadas con las preguntas ya expuestas para este caso y son:

```
(Gusta_mat Si/No/Fin)
(Es_trabajador Mucho/Normal/Poco/Fin)
(Gusta_prog Si/No/No se/Fin)
(Gusta_hardw Si/No/No se/Fin)
(Nota Alta/Media/Baja)
```

Las pocas observaciones a hacer es que no es necesario responder a todas las preguntas para obtener el consejo. En otras palabras no es necesario asertar todas estas variables de entrada al sistema para obtener el consejo. El flujo de las preguntas que se realizan y por consiguiente de los hechos que se asertan al sistema depende del árbol de decisión. Podemos observar esto en la Imagen 1. Esto hace amena la labor del usuario. Asimismo el hecho (Nota Alta/Media/Baja) es indirectamente introducido por el usuario ya que éste introduce su calificación media (Calificacion\_media ?num) la cual el sistema transforma.

Para el caso de **Asignaturas** las variables de entrada a considerar están relacionadas con las preguntas ya expuestas para este caso y son:

```
(OpcionElegida CODIGO)
(AsigDeseadas 1 ENTERO)
(AsigDeseadas 2 ENTERO)
(Character Teorico/Practico/Fin)
(Dificultad Facil/Difícil/Fin)
(Trabajo Mucho/Poco/Fin)
```

Comenzamos destacando que **CODIGO** es el código de una asignatura (las letras iniciales) y **ENTERO**  $\in [0, 5]$ . Del mismo modo el usuario no tiene que introducir las asignaturas una por una. Las introduce separadas por espacios, como por ejemplo **SO FR ED** y éstas se asertan como (**OpcionesElegidas SO FR ED**). El sistema se encargará de realizar el procedimiento para obtener los hechos que finalmente deseamos tener en el conocimiento.

Nuestro interés no reside en el número de créditos a matricular sino en el número de asignaturas porque todas ellas son de 6 créditos. El hecho **AsigDeseadas** guarda el número de asignaturas que el usuario desea matricular en cada cuatrimestre.

### 3.2. Variables de salida del problema

Las variables de salida del sistema son los consejos que se dan al usuario para asesorarle. En este caso no hace falta distinguir entre ambos módulos porque la finalidad es la misma. Lógicamente los consejos se representan en el sistema por igual. La representación es:

(Consejo ?rama/asignatura ?texto ?experto)

donde la primera componente indica la rama o asignatura (en caso de ser asignatura se usa el código), la siguiente componente es el texto con el motivo de ese consejo y por último el experto que realiza el consejo.

### 3.3. Conocimiento global del sistema

En este apartado se van a comentar aquellos hechos que son parte del sistema al comienzo y que no son las variables de entrada. Este es conocimiento que poseen y cargan directamente.

Comenzamos por aquellos hechos que se asertan al comienzo del módulo **RAMAS**:

```
(Rama Computacion_y_Sistemas_Inteligentes)
(Rama Ingenieria_del_Software)
(Rama Ingenieria_de_Computadores)
(Rama Sistemas_de_Informacion)
(Rama Tecnologias_de_la_Informacion)
```

El único conocimiento que se necesita es el de las Ramas. Dicha información sumada a la de las variables de entradas es la necesaria para el asesoramiento. Comenzamos por aquellos hechos que se asertan al comienzo del módulo **ASIGNATURAS**:

```
(Asignatura Estructura_de_Datos ED 2 1 Practica Dificil Mucho)
(Asignatura Estructura_de_Computadores EC 2 1 Teorica Facil Poco)
(Asignatura Sistemas_Concurrentes_y_Distribuidos SCD 2 1 Practica Facil Poco)
(Asignatura Sistemas_Operativos SO 2 1 Teorica Dificil Mucho)
(Asignatura Programacion_y_Disenio_Orientado_a_Objeto PODO 2 1
  Practica Dificil Mucho)

(Asignatura Algoritmica ALG 2 2 Practica Facil Poco)
(Asignatura Arquitectura_de_Computadores AC 2 2 Teorica Facil Mucho)
(Asignatura Fundamentos_de_Bases_de_Datos FBD 2 2 Practica Facil Poco)
(Asignatura Fundamentos_de_Ingenieria_del_Software FIS 2 2 Teorica Facil Poco)
(Asignatura Inteligencia_Artificial IA 2 2 Practica Dificil Mucho)

(Asignatura Diseno_y_Desarrollo_de_Sistemas_de_la_Informacion DDSI 3 1
  Practica Facil Poco)
(Asignatura Fundamentos_de_Redes FR 3 1 Teorica Dificil Mucho)
(Asignatura Informatica_Grafica IG 3 1 Practica Dificil Mucho)
(Asignatura Ingenieria_de_Servidores ISE 3 1 Teorica Dificil Mucho)
(Asignatura Modelos_de_Computacion MC 3 1 Teorica Facil Mucho)

(SoloConsideradas FALSE)
(ListoParaAconsejar FALSE)

(AsigDeseadas 2 -1)
(AsigDeseadas 1 -1)

(ContarAsignaturas 2)
(ContarAsignaturas 1)

(ContadorAconsejadas 2 0)
(ContadorAconsejadas 1 0)

(ContadorConsejosSeguros 2 0)
(ContadorConsejosSeguros 1 0)
```

Como vemos la información de las asignaturas guarda el nombre completo de la asignatura, su código (iniciales), el curso, el cuatrimestre, si tiene un carácter práctico o teórico,

si es difícil o fácil y si tiene mucho o poco trabajo. Sólo se han incluido asignaturas de segundo y tercero del Grado en I. Informática. Para incluir más asignaturas sólo es necesario meterlas en el conocimiento al lado de éstas. Explico el resto de hechos del sistema:

- **SoloConsideradas:** El valor que le acompaña es booleano e indica si las asignaturas del sistema son las que el usuario quiere que se consideren. El sistema elimina en primer lugar todas las asignaturas que no tienen un hecho **OpcionElegida** asociado. Posteriormente se borran también los hechos **OpcionElegida** dejando en el sistema únicamente las asignaturas consideradas por el usuario y estableciendo en **TRUE** el valor de **SoloConsideradas**.
- **ListoParaAconsejar:** En el momento en el este hecho esté activado el sistema imprime por pantalla el consejo al usuario. Por ejemplo cuando el usuario responde **Fin** entonces el valor de **ListoParaAconsejar** se establece a **TRUE**. Por defecto es **FALSE**.
- **AsigDeseadas:** Hay un hecho de este tipo para cada cuatrimestre e indica el número de asignaturas que el alumno desea matricular en ese cuatrimestre.
- **ContarAsignaturas:** Mantiene un conteo actualizado en tiempo real del número de asignaturas en el sistema. Es decir, el total de las opciones que tiene el usuario para elegir. Si se diera la condición de que el número de asignaturas deseadas es mayor que el número de asignaturas existentes entonces se asertaría (**ListoParaAconsejar TRUE**).
- **ContadorAconsejadas:** Existe un hecho para cada cuatrimestre y básicamente es un contador del número de asignaturas que ya se le han aconsejado al usuario, imprimidas por pantalla. En ningún caso el sistema aconsejará más asignaturas de las indicadas ya que el diseño que se pretendía era así.
- **ContadorConsejosSeguros:** Mantiene un conteo del número de consejos que tienen una razón de peso para ser aconsejados ya que de manera inicial los consejos son por defecto y no existe un motivo explícito. En el instante en el que se verifique que el número de consejos seguros sea igual al de los deseados ya podemos finalizar usando (**ListoParaAconsejar TRUE**).

### 3.4. Especificación de los módulos desarrollados

La estructura de los módulos es la siguiente. Tal y como ya se ha anunciado el módulo principal es el módulo **MAIN** que gestiona la bienvenida al sistema y los transposos entre los otros dos módulos existentes: **RAMAS** y **ASIGNATURAS**. Sus definiciones son las siguientes:

```
(defmodule MAIN (export ?ALL))

(def facts MAIN::Main
  (Elige 0)
)

(defmodule RAMAS (import MAIN ?ALL))

(defmodule ASIGNATURAS (import MAIN ?ALL))
```



Observamos que el resto de módulos importan información esencial de **MAIN** como es **initial-fact** (esto es obligatorio) pero además importan otras cosas como el hecho **Elige**, el cual es importante para saber los asesoramientos que el usuario ha recibido en dicha sesión.

Respecto a la tarea que realiza cada módulo ya ha sido suficientemente explicada y detallada. La tarea es aconsejar **Ramas** o **Asignaturas** como los nombres de los módulos ilustran. El conocimiento que usan y deducen está explicado extensamente en la Subsección 3.3.

### 3.5. Hechos y las reglas de cada módulo

Los hechos del sistema ya se han comentado en su práctica totalidad. Conocer brevemente las reglas del sistema es importante para su correcto entendimiento. El módulo **MAIN** sólo tiene una regla.

- **Da\_bienvenida**: Recibe al usuario y explica para qué sirve el sistema a la vez que pide elegir entre **Ramas** y **Asignaturas**.

Las reglas del módulo **RAMAS** son:

- **Da\_bienvenida\_Ramas**: Recibe al usuario en este módulo explicándole las posibles **Ramas**.
- **Gusta\_mat**: Pregunta al usuario si le gustan las matemáticas.
- **Gusta\_mat\_check**: Comprueba que la respuesta introducida sea válida.
- **Es\_trabajador**: Pregunta al usuario si se considera trabajador.
- **Es\_trabajador\_check**: Comprueba que la respuesta introducida sea válida.
- **Gusta\_prog**: Pregunta al usuario si le gusta programar.
- **Gusta\_prog\_check**: Comprueba que la respuesta introducida sea válida.
- **Gusta\_hardw**: Pregunta al usuario si le gusta el hardware.
- **Gusta\_hardw\_check**: Comprueba que la respuesta introducida sea válida.
- **Trabajar**: Pregunta al usuario dónde quiere trabajar.
- **Nota**: Pregunta al usuario su calificación media de expediente comprobando que lo introducido sea un número.
- **Nota\_check\_interval**: Comprueba que la respuesta introducida esté en el intervalo  $[5,10]$  porque sólo eso es posible.
- **Nota\_conversion\_alta**: Convierte la nota a Alta.
- **Nota\_conversion\_media**: Convierte la nota a Media.

- `Nota_conversion_baja`: Convierte la nota a Baja.
- `imprime_consejo`: imprime el consejo acerca de la Rama

Los siguientes hechos introduce un `Consejo` de esa Rama en el sistema y lo hacen cuando se cumplen unos ciertos antecedentes. Algunas veces antecedentes distintos puedan dar lugar a aconsejar una misma Rama tal y como vemos en el árbol de decisión de la Imagen 1. Estas reglas son: `Rama_CSI`, `Rama_IS`, `Rama_IS2`, `Rama_IC`, `Rama_IC2`, `Rama_SI`, `Rama_TI`, `Rama_TI_No_se`, `Rama_CSI_No_se` y `Gusta_hardw_No_se`.

Cuando el usuario introduce `Fin` se debe dar un consejo con el conocimiento ya existente del sistema y del usuario que en algunos casos será mayor y en otros menos. Las reglas que se encargan de esto son: `Fin_1`, `Fin_3`, `Fin_4` y `Fin_5`.

Finalizamos el apartado explicando las reglas del módulo `ASIGNATURAS`. Veamos:

- `Da_bienvenida_Asignaturas`: Recibe al usuario en este modulo.
- `Imprime_Asignaturas_2`: Imprime las asignaturas de segundo.
- `Imprime_Asignaturas_3`: Imprime las asignaturas de tercero.
- `imprime_consejo_seguro`: Imprime aquellos consejos seguros.
- `imprime_consejo_por_defecto`: Imprime consejos que son por defecto.
- `contar`: Inicia el contador de `AsigExistentes`.
- `contar_hecho`: Cuenta `AsigExistentes`.
- `borrar1`: Borra variables necesarias en el proceso de conteo.
- `borrar2`: Borra variables necesarias en el proceso de conteo.
- `Indique_Asignaturas`: Solicita las asignaturas que desea que se consideren.
- `Opciones_Elegidas`: Transforma el hecho (`Opciones_Elegidas $?vector`) en un vector de hechos del tipo (`OpcionElegida ?op`).
- `Check_Opcion_Elegida`: Comprueba que las opciones elegidas son asignaturas existentes y válidas.
- `borrar_op`: Borra el hecho (`opcionesElegidas`) ya que el vector está vacío.
- `Borrar_Asig_No_Consideradas`: Borra del sistema las asignaturas no consideradas por el usuario.
- `Pedir_Asig`: Pide la cantidad de asignaturas que el usuario desea matricular en un cuatrimestre comprobando que es un valor entero.
- `Check_Asig`: Debido a la limitación de las asignaturas que el sistema conoce no le deja matricularse de más de 5 asignaturas por cuatrimestre.
- `Cambiar_Solo_Consideradas`: Establece el verdadero el hecho `SoloConsideradas` retractando el existente y asertando (`SoloConsideradas TRUE`).

- **Borrar\_Opcion\_Elegida:** Cuando existe (`SoloConsideradas TRUE`) las asignaturas del sistema son las que el usuario quiere por lo que se pueden borrar las opciones elegidas `OpcionElegida`.
- **Introduce\_Consejo:** Crea consejos por defecto para las asignaturas.
- **Fin\_cuatri:** Cuando el número de consejos de un cuatrimestre alcanza los deseados se introduce en el sistema el hecho (`FinCuatri CUATRIMESTRE`).
- **Fin\_Consejo:** Cuando los cuatrimestres finalizan, (`ListoParaAconsejar TRUE`).
- **Siguiente1:** Pregunta si prefiere asignaturas teóricas o prácticas.
- **Teorica\_Practica:** Razona consejos en función de la repuesta.
- **Siguiente\_2:** Pregunta si prefiere asignaturas difíciles o fáciles.
- **Facil\_Dificil:** Razona consejos en función de la repuesta.
- **Siguiente3:** Pregunta si prefiere asignaturas con mucho o poco trabajo.
- **Mucho\_Poco:** Razona consejos en función de la repuesta.
- **Ir\_Ramas:** Reconduce el flujo al módulo `RAMAS` si el usuario lo desea.

## 4. Breve manual de uso del sistema

El archivo implementado se llama `practica_final.clp` por lo que las primeras ordenes a introducir como ya sabemos son:

```
(load ruta/practica_final.clp), (reset) y (run).
```

En este momento comienza la ejecución. El sistema da la bienvenida y ofrece dos posibilidades **Ramas** y **Asignaturas**.

- **Ramas.** Si se elige esta opción se formularan las preguntas indicadas en la sección 1. En breves preguntas el sistema tendrá un consejo muy elaborado y se lo ofrecerá al usuario argumentando la decisión. En caso de que se den respuestas de información parcial se trabajará con ellas o incluso se harán preguntas relacionadas con la finalidad de que el sistema deduzca este conocimiento. Por ejemplo si el usuario no sabe si le gusta el hardware se le puede preguntar si le gustaron asignaturas como FFT y EC. Asimismo, se puede responder `Fin` lo cual propociona un Consejo pero con menor elaboración.
- **Asignaturas.** La preguntas son las indicadas en la sección 1. La primera de ellas es las asignaturas (de las ofrecidas en el menú) que quiere que se consideren en el asesoramiento. Además, se realiza distinción por cuatrimestres para ofrecer mayor precisión y realismo al aconsejar. A partir de ahí hay un máximo de tres preguntas para conocer los gustos del alumno y llevar a cabo la recomendación.

Una vez que termine el consejo de uno de estos dos campos el sistema ofrece la posibilidad de recibir asesoramiento del otro. En caso negativo finaliza la ejecución y en el positivo después de aconsejar ya no preguntará si quiere consejo del otro pues el sistema sabe que ya recibió ese asesoramiento.