

# **INTEL·LIGÈNCIA ARTIFICIAL**

## **SISTEMAS BASADOS en el CONOCIMIENTO**

DAVID GARCÍA, DAVID HIDALGO, JOAN MARC COLL

18 de diciembre 2023



# Índice

<b>1 Introducción</b>	<b>4</b>
<b>2 Problema</b>	<b>5</b>
2.1 Descripción del problema	5
2.2 Viabilidad de la solución	5
2.3 Fuentes de conocimiento	6
2.4 Objetivos	7
<b>3 Conceptualización</b>	<b>8</b>
3.1 Conceptualización del problema	8
3.2 Elementos del dominio	8
3.3 Problemas y subproblemas	8
3.3.1 Recogida de datos del lector.	9
3.3.2 Análisis de las preferencias del lector.	9
3.3.3 Recomendación de tres libros al lector.	9
3.4 Ejemplos de conocimiento experto extraído del dominio.	10
3.5 Proceso de resolución de problemas y subproblemas.	10
<b>4 Formalización</b>	<b>11</b>
4.1 Desarrollo de la ontología	11
4.1.1 Representación de clases y jerarquía	11
4.1.2 Definición de clases y relaciones	13
4.2 Metodología de resolución. Clasificación heurística.	16
4.2.1 Abstracción de los datos	16
4.2.2 Asociación heurística.	16
4.2.3 Refinamiento	16
<b>5 Implementación</b>	<b>18</b>
5.1 Representación de la ontología	18
5.2 Resolución del problema	18
5.2.1 input	18
5.2.2 inferencia-datos	19
5.2.3 ponderar-libros	19
5.2.4 recomendar-libros	19
5.3 Proceso de resolución	20
5.3.1 Reglas	20
5.3.2 Funciones	21
5.4 Desarrollo de prototipos	22

<b>6. Juegos de prueba</b> .....	<b>23</b>
6.1 Recomendación de autores extranjeros .....	23
6.2 Recomendación de autores nacionales .....	24
6.3 Recomendación por género requerido .....	26
6.4 Recomendación por época requerida .....	27
6.5 Requerimientos muy exigentes .....	29
6.6 Recomendación de best-sellers para completar .....	31
<b>7. Planificación de tareas</b> .....	<b>33</b>

# **1 Introducción**

Realización de la segunda práctica de laboratorio, donde nuestro objetivo es resolver un problema de recomendación de libros mediante un sistema basado en el conocimiento.

Concretamente, trabajaremos con el programa Protégé y nos familiarizaremos con CLIPS, que es un lenguaje de programación especializado y utilizado en construcción de sistemas expertos. Nos servirá para representar conocimientos en forma de reglas de producción y para realizar razonamiento basado en reglas en SBC.

Después de pensar en varias implementaciones de ontologías y modificarlas en numerosas ocasiones, haber pensado con ahínco las reglas a implementar según las preferencias del lector que nos va a realizar la recomendación, así como la estructuración de un código legible y modular, tenemos el gusto de presentar nuestro sistema experto basado en el conocimiento.

## **2 Problema**

### **2.1 Descripción del problema**

El problema consiste en recomendar tres libros a un lector a partir de un sistema basado en el conocimiento. Lo que buscamos es acercarnos al lector para poder hacer una recomendación mucho más precisa sin obligatoriamente tener que seguir los estándares de la gente; es decir, personalizarla mucho más, ya sea obteniendo información de sus gustos, preferencias o requerimientos.

Aún así, para hacer la recomendación nos centraremos en las novelas de ficción. No obstante, para estas tenemos diferentes géneros, lo que conlleva a que nuestro abanico de recomendación pueda ser mucho más variado.

Lo primero que tendremos que tener en cuenta será el perfil del lector. Mediante preguntas, podremos obtener información valiosa tales como el lugar donde va a efectuar la lectura, el momento que dispone para leer, su frecuencia de lectura, si observa valoraciones, o si prefiere autores extranjeros, los libros que ha adquirido para saber qué autores son sus preferidos o los géneros que prefiere, por ejemplo. Así luego podremos concretar más al recomendar ciertos libros.

Por otra parte, también deberemos tener en consideración los libros. Éstos pueden ser de uno o muchos géneros (p.ej: *El Libro de Todos Los Géneros*, donde cada capítulo está ambientado en un género), escritos por un autor, tienen una valoración de los usuarios que lo han leído, su complejidad, la portabilidad, etc. En cuanto a su clasificación, será muy importante distinguirlos entre libros populares y aquellos que no son tan conocidos.

Además, también hay autores, que se caracterizan por su nacionalidad y época en la que escriben, el género especializado en el que están, los libros que ha escrito, y su nombre. Por último, pero no menos importante, tenemos a los géneros. Estos se caracterizan por su nombre.

Si después de toda la información no es suficiente para dar una recomendación completa de tres libros, se deberá recomendar alguno de los ejemplares más vendidos del momento que se acerquen más a las características del lector que se han obtenido.

### **2.2 Viabilidad de la solución**

Nuestra idea de utilizar el sistema basado en el conocimiento es clara. Queremos conceptualizar la información del problema para así poder recomendar libros de una manera más eficiente y precisa. Por ende, no utilizar un SBC nos estaría complicando

mucho más nuestra finalidad, ya que no tendríamos constancia del perfil del lector ni de sus libros leídos para poder recomendar libros ajustados de entre todos los del sistema.

Aplicando reglas y restricciones podremos obtener las preferencias y gustos del lector, cosa que nos ayudará a mejorar la calidad de la recomendación, ya que podremos sugerir libros similares a sus respectivas lecturas.

### **2.3 Fuentes de conocimiento**

Para poder hacer que nuestro sistema experto funcione correctamente, debemos nutrirlo de fuentes de conocimiento tales como libros de diferentes géneros, épocas, autores, determinando su portabilidad y su popularidad, entre otras, ya que tendremos más variedad para recomendar y el sistema podrá escoger más ejemplares sobre esos requerimientos y preferencias.

No obstante, no nos podemos olvidar del perfil del lector. También será vital ese conocimiento para poder saber qué géneros son los que prefiere en base a los libros que ha leído, o determinar si le gustan libros de valoraciones altas porque todos los que ha leído superan la valoración en más de 4 puntos, o si le gustan los libros con autores extranjeros ya que más de la mitad de sus ejemplares son de autores de nacionalidad diferente. También, nos será útil para determinar si los libros a recomendar deberán ser de mayor/menor complejidad dependiendo de su edad, frecuencia de lectura anual, lugar de lectura y momento de lectura.

Lo primero de todo, deberemos tener esas instancias en nuestro sistema. Para ello, hemos hecho una búsqueda por internet sobre diferentes libros, teniendo en cuenta su popularidad, género, época en la que el autor de su libro ha escrito, si es complejo de leer, entre otras. Nos interesará tener cuanta más variedad mejor, ya que podremos hacer un juego de pruebas más rico a la hora de testear nuestro programa.

Después de obtener la información relevante de los primeros libros encontrados, decidimos utilizar la herramienta de Chat GPT para que nos generase muchas más instancias de libros variados con sus respectivas particularidades. Algunos de ellos los hemos editado nosotros también; es decir, cambiado algún parámetro como por ejemplo, la complejidad, buscando así casos extremos a la hora de la recomendación.

Lo mismo para los autores, el sistema sabrá a partir de los libros que ha leído el género en el que están especializados, o la época en la que escriben.

Dentro del marco del programa, a medida que obtenemos la información del lector y sus preferencias, tendremos el conocimiento de los libros que son más adherentes a las especificaciones. Esto se llevará a cabo mediante una ponderación de puntuación para cada libro. De esta forma, recomendaremos los libros con mayor puntuación, que en el cómputo global indican que son los más recomendables.

## **2.4 Objetivos**

Para que el sistema experto realice una buena recomendación, hemos planteado los siguientes objetivos. Sabemos que, si los seguimos ceñidamente, llegaremos a encontrar una solución donde los libros resultantes a recomendar sean los especificados por el lector.

- I. Preguntar al usuario sobre sus requerimientos. No es lo mismo querer que preferir. Por tanto, le haremos un mínimo de preguntas, pues no queremos que sea un interrogatorio.
- II. Obtener muchas más peculiaridades del lector, sobre todo aquellas que no hagan falta preguntar y se sepan a partir del conocimiento extraído de su perfil.
- III. Potenciar libros a recomendar o descartar algunos otros a partir de la información que tengamos extraída y preguntada.
- IV. Hacer una recomendación adecuada a sus requerimientos y preferencias, lo más ceñida posible, si es que contamos con ese tipo de libros entre nuestras instancias.
- V. En caso de no ser posible dicha recomendación, mostrarle tres libros (o los necesarios para una recomendación completa) populares en cuanto a sus ventas, o también denominados best-sellers,, que más se ajusten a sus preferencias.

## 3 Conceptualización

### 3.1 Conceptualización del problema

Partimos de un sistema de recomendación diseñado por nosotros, cuya base de datos está formada por un conjunto de libros con sus respectivas características, y un conjunto de lectores registrados.

### 3.2 Elementos del dominio

El dominio de nuestro problema se basará en las características indicadas en el enunciado de la práctica. También añadiremos conceptos que consideramos relevantes para la recomendación de libros que se ajusten a los gustos del lector.

Disponemos de dos grandes clases:

#### libro

Contendrá los elementos que le recomendaremos al lector según sus preferencias, se identificarán por su nombre y cada libro tendrá asociado al autor por el cual fue escrito, junto a los géneros que contiene, a parte de varias características específicas de cada uno como su valoración, complejidad, popularidad y portabilidad.

#### autor

Contendrá los autores de los libros de nuestro dominio, estos se identificarán por su nombre y tendrán características como su época o los géneros literarios en los que se especializan.

#### Características de un género

Contendrá los géneros de los libros de nuestro dominio, estos se identificarán por su nombre y tendrán características como su época o los géneros literarios en los que se especializan.

#### Características de un lector

Contendrá los lectores registrados en nuestra base de datos, cada uno se identificará por su nombre, y tendrá un conjunto de características como su edad o nacionalidad.

### 3.3 Problemas y subproblemas

Nuestro problema consiste en encontrar los tres libros que más se adecuen a los gustos del lector, para ello lo hemos dividido en los siguientes subproblemas.



### **3.3.1 Recogida de datos del lector**

Al inicio el sistema pedirá que el usuario especifique cuál de los lectores es de los que tenemos registrados en nuestra base de datos. Seguidamente cada lector de nuestro sistema podrá realizar consultas para encontrar libros especificando ciertos requerimientos. Por cada consulta se les preguntará por su frecuencia de lectura anual, momento y lugar preferidos de lectura, sus preferencias respecto a la popularidad de los libros que quiere leer, y si quiere que su libro sea de alguna época o pertenezca a una serie de géneros en específico.

### **3.3.2 Análisis de las preferencias del lector.**

El sistema, para traducir toda la información que tiene, ponderará cada libro según los siguientes criterios:

- 1) La complejidad de la lectura que el lector podrá asumir, la cual se determinará a partir del lugar en el que vaya a leer, el momento del día en el que lo hará, su edad, y cuán frecuentemente lee.
- 2) Su preferencia de géneros, la cual se determinará a partir del conjunto de libros que el lector haya adquirido en un pasado, y a partir de los géneros en los que se especialicen sus autores preferidos.
- 3) Su preferencia en cuanto a autores, que se determinará en base a los autores de los libros que el lector haya adquirido, también a partir de la época literaria a la cual pertenecen estos autores, y a su nacionalidad.
- 4) Si requiere que el libro sea portable según el lugar de lectura en el que prevé que va a leer.
- 5) Si el lector le da importancia a las valoraciones de los libros que ha adquirido.
- 6) Si el lector le da importancia a la popularidad de los libros que ha adquirido.

### **3.3.3 Recomendación de tres libros al lector**

Una vez ponderados todos los libros ya se sabe cuales son aquellos con mayor probabilidad de gustarle al lector, y ahora se filtra según las exigencias en cuanto a géneros, épocas y popularidad que haya demandado el lector inicialmente. De esta manera, los tres libros con mayor puntuación después del filtrado serán los más interesantes para el lector.

En caso que después del filtrado haya menos de tres libros para recomendar que cumplan las exigencias del lector, se informará al usuario de ello y se recomendará aquellos que sí cumplan las exigencias, siendo los restantes los

bestsellers que más se adecuen a las preferencias del lector según su puntuación.

### **3.4 Ejemplos de conocimiento experto extraído del dominio.**

Para poder extraer los principios generales que podrán ser aplicados a la recomendación de un libro desde el punto de vista de un experto analizaremos los datos que nos ha proporcionado el lector, y las características de los libros que adquirió en un pasado.

- En función de si el usuario tiene una edad corta, lee con poca frecuencia, lee en un momento del día en el cual la actividad cerebral es baja como en la noche, o lee en un lugar poco cómodo; se intentará asignar libros de una complejidad más baja.

- Si el usuario leerá en un lugar poco cómodo, se intentará asignar libros que sean portables.

- Si los libros que ha adquirido el usuario tienen una valoración media por encima del 4, intentaremos asignar libros con valoraciones altas.

- Si la mayoría de los libros que ha adquirido el usuario son de autores extranjeros, se intentará asignar libros cuyos autores sean de una nacionalidad distinta a la del lector.

- En función de la cantidad de veces que el usuario haya leído un género, se le dará más prioridad a estos.

- En función de la cantidad de veces que el usuario haya leído un autor, se le dará más prioridad a estos.

- Si la colección de libros de un usuario está formada por al menos una sexta parte de libros del mismo autor, daremos más prioridad al género en el que se especialice este autor.

- En función de la cantidad de veces que el usuario haya leído autores de una época, se le dará más prioridad a autores de la misma época.

- En caso de que la gran mayoría de autores que haya leído el usuario sean de una nacionalidad distinta a la suya, se le dará más prioridad a autores extranjeros.

### **3.5 Proceso de resolución de problemas y subproblemas.**

Partimos de la información proporcionada por el usuario sabiendo que nos interesa resolver el problema haciendo uso de un SBC. A partir de aquí daremos una prioridad absoluta a las exigencias que haya realizado el usuario, ordenando el conjunto de libros que cumplan con estas exigencias a partir de la abstracción de los datos que nos haya proporcionado el usuario, asociándole una metaheurística a cada uno de estos datos abstractos.

## **4 Formalización**

### **4.1 Desarrollo de la ontología**

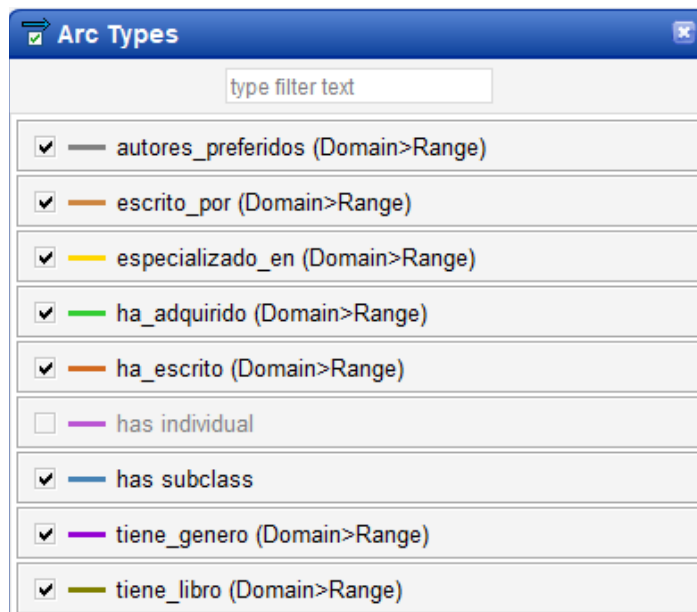
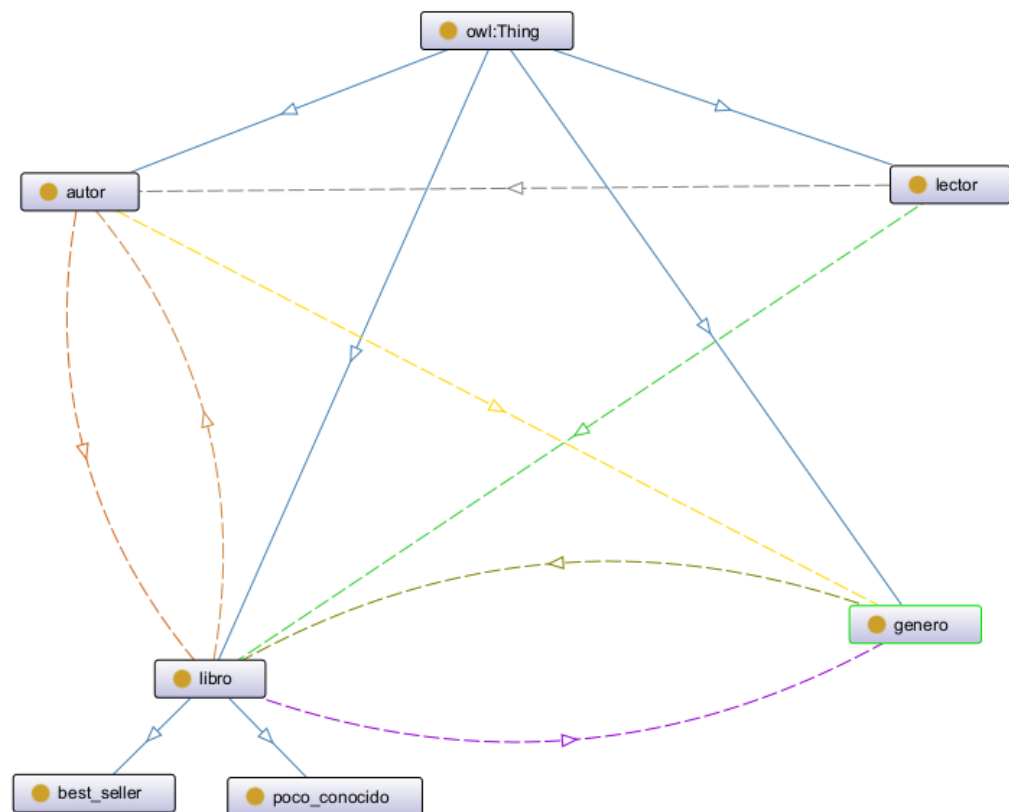
Una vez definidos los elementos del dominio, debemos trasladarlos formalmente a una ontología. Para ello, es necesario crear las clases necesarias para representar estos elementos y especificar qué atributos las conforman. Además, hemos de definir las relaciones que existen entre estas diferentes clases.

Para llevar a cabo todo este proceso, es realmente importante preguntarse cuándo es necesario crear una nueva clase para representar un elemento del dominio, o si simplemente es suficiente con añadirlo como un atributo de una clase ya existente. Otra cosa a tener en cuenta es mantener la ontología equilibrada respecto a sus subclases, es decir, evitar que existan ramas de subclases descompensadas respecto a otras de su mismo nivel. Para evitar este tipo de problemas, hemos optado por un diseño de ontología simple, con pocas clases, simplificando también la posterior programación de reglas.

La ontología resultante debe ser capaz de representar formal y explícitamente todas las categorías y relaciones pertenecientes al dominio, para así mejorar la comprensión y el procesamiento de la información por parte de las máquinas, estandarizar el conocimiento y facilitar la inferencia lógica.

#### **4.1.1 Representación de clases y jerarquía**

El siguiente esquema pretende representar todas las clases de la ontología completa y las relaciones que se dan entre ellas.



A continuación profundizamos en cada una de las clases y presentamos sus distintos atributos y relaciones:

#### 4.1.2 Definición de clases y relaciones

##### Autor

Esta es la clase que contiene toda la información referente a un autor. Sus atributos nos permiten identificarlo e inferir información nueva, como por ejemplo definir en qué género está especializado.

Autor	
epoca	Integer
nacionalidad	String
nombre	String
ha_escrito	Instance* Libro
especializado_en	Instance* Genero

- **epoca**: Entero que representa el índice de la época a la que pertenece el autor.
- **nacionalidad**: String que representa la nacionalidad del autor.
- **nombre**: String que representa el nombre del autor.
- **ha\_escrito**: Relación que apunta a los libros que ha escrito el autor.
- **especializado\_en**: Relación que apunta a los géneros en los que está especializado el autor.

##### Genero

Clase que representa un género literario. La información que contiene es la justa y necesaria para poder identificar cada uno de los géneros y que libros existe de ese mismo género.

Genero	
nombre	String
tiene_libro	Instance* Libro

- **nombre**: String que representa el nombre del género literario.
- **tiene\_libro**: Relación que apunta a los libros que pertenecen al género.

## Lector

La clase lector es una de las clases más importantes en la ontología. Esta contiene toda la información referente al lector que solicita una recomendación, incluyendo tanto información que él proporciona como información que el sistema infiere a partir de la entrada.

Lector	
complejidad	Float
edad	Integer
epoca_quiere	Integer
epocas_prefiere	Integer
frec_lectura_anual	Integer
generos_prefiere	Integer
generos_quiere	Integer
gustan_autores_extranjeros	Boolean
lugar_lectura	Integer
momento_lectura	Integer
nacionalidad	String
nombre	String
observa_valoraciones	Boolean
preferencias_best_sellers	Integer
autores_preferidos	Instance* Autor
ha_adquirido	Instance* Libro

- **complejidad:** Número en coma flotante que indica la complejidad media de todos los libros que ha adquirido el lector.
- **edad:** Entero que representa la edad del lector.
- **epoca\_quiere:** Entero que representa el índice de la época que el lector exige en su recomendación.
- **epocas\_prefiere:** Enteros que indican el grado de preferencia que tiene el lector por cada una de las épocas.
- **frec\_lectura\_anual:** Entero que representa la cantidad de libros que ha leído el lector en el último año.
- **generos\_prefiere:** Enteros que indican el grado de preferencia que tiene el lector por cada una de los géneros.
- **generos\_quiere:** Entero que representa el índice del género que el lector exige en su recomendación.
- **gustan\_autores\_extranjeros:** Booleano que indica si el autor tiene especial interés en autores que no són de su misma nacionalidad.

- **lugar\_lectura**: Entero que representa el índice del lugar en el que el lector acostumbra a leer.
- **momento\_lectura**: Entero que representa el índice del momento en el que el lector acostumbra a leer.
- **nacionalidad**: String que representa la nacionalidad del lector.
- **nombre**: String que representa el nombre del lector.
- **observa\_valoraciones**: Booleano que indica si el lector tiene especial interés en libros que están bien valorados.
- **preferencias\_best\_sellers**: Entero que indica si el lector prefiere un bestseller o libro poco conocido, o si no le importa.
- **autores\_preferidos**: Relación que apunta a los autores que son considerados preferidos por el sistema en base a los libros que ha adquirido.
- **ha\_adquirido**: Relación que apunta a los libros que el lector ha adquirido en algún momento.

## Libro

Clase esencial que representa toda la información de un libro y cómo se relaciona con el resto de clases. De ella cuelgan dos subclases, `best_seller` y `poco_conocido`. Esta subclasificación es completa y disjunta, y nos permite diferenciar los libros muy populares de aquellos que no lo son tanto.

Libro	
titulo_libro	String
complejidad	Float
npaginas	Integer
valoracion	Float
portable	Boolean
escrito_por	Instance* Autor
tiene_genero	Instance* Genero

- **titulo\_libro**: String que representa el título del libro.
- **complejidad**: Float que representa la complejidad del libro.
- **npaginas**: Entero que indica el número de páginas del libro.
- **valoracion**: Float que representa la valoración que recibe el libro en este momento.
- **portable**: Booleano que indica si las características del libro permiten un mínimo de portabilidad.

- **escrito\_por**: Relación que apunta al autor del libro.
- **tiene\_genero**: Relación que apunta al género del libro.

## **4.2 Metodología de resolución. Clasificación heurística.**

La tarea de recomendar libros mediante un Sistema Basado en el Conocimiento involucra principalmente tareas de análisis. Esto se debe a que el sistema necesita analizar la información de entrada, que puede incluir las preferencias del usuario, datos demográficos, historial de lecturas anteriores, calificaciones, etc. Este análisis ayuda al sistema a comprender las necesidades y preferencias del usuario. Además de esto, el SBC debe analizar la base de conocimientos, que contiene información sobre los libros disponibles, sus géneros, autores, temáticas, estilos de escritura, etc. Esto implica evaluar y comparar las características de los libros para determinar su idoneidad para el usuario. Es por eso que el método de resolución escogido es la clasificación heurística.

### **4.2.1 Abstracción de los datos**

La abstracción de datos es un paso fundamental en el diseño de sistemas de recomendación. En nuestro caso concreto, consiste en identificar y seleccionar las características más relevantes de los datos proporcionados por el usuario y convertirlos en hechos generales. De esta forma la complejidad de los datos se simplifica permitiendo un procesamiento más eficiente y efectivo. Un ejemplo claro podría ser deducir si al lector le interesan o no los autores extranjeros a partir de los libros que ha adquirido anteriormente.

### **4.2.2 Asociación heurística**

La asociación heurística implica el uso de reglas heurísticas para relacionar los datos abstractos obtenidos y encontrar una solución abstracta válida. Para conseguirlo, hemos desarrollado un sistema de puntuación que, aplicando reglas sobre los datos abstraídos, otorga más o menos puntuación a cada uno de los libros. Aquellos libros que no cumplan las exigencias del lector no son considerados.

### **4.2.3 Refinamiento**

Finalmente, en el proceso de refinamiento convertimos la solución abstracta obtenida anteriormente en una solución concreta. Para ello, se escogen los tres libros con mayor puntaje y se presentan con todas sus características al lector,



añadiendo también información que el sistema ha inferido, para así justificar de cierta manera la recomendación.

## 5 Implementación

### 5.1 Representación de la ontología

En cuanto a la representación de la ontología, simplemente deberemos hacer la conversión de .ttl en Protégé a .clp en CLIPS. Sabemos que en la ontología está representado todo el conocimiento, pero una vez desde clips, modificaremos aquellos atributos de las clases que no sean fijos, tales como las frecuencias de lectura anuales, los lugares y momentos de lectura, géneros que prefiera, o incluso la época y si requiere un best-seller. Dejaremos como atributos de las clases aquellos que sean fijos, como por ejemplo el nombre, la edad, el número de páginas de un libro en concreto; sabemos que estos no cambiarán en el tiempo.

Para hacer dicha separación, necesitamos plasmar ese conocimiento en deftemplates, los cuales crearemos tres; uno para los que necesitemos para preguntar los datos de entrada, otro para representar los conocimientos de los datos inferidos, y por último, un deftemplate para guardar los tres libros finales a recomendar.

En el deftemplate de los datos de entrada tendremos un slot para el lector que desea que le hagamos la recomendación, la frecuencia, momento y lugar de lectura, la época y popularidad que requiere y un multi slot de los géneros que requiere.

Para el segundo deftemplate, el de los datos inferidos, tendremos un slot para la complejidad de la recomendación; que nos servirá para recomendar libros parecidos en cuanto a dificultad, otro slot para determinar si preferiría que los libros fueran portables, y finalmente dos slots más para saber si le gustan autores extranjeros o si observa valoraciones.

### 5.2 Resolución del problema

Para resolver el problema, hemos dividido éste en 4 módulos.

Uno nos servirá para preguntarle al usuario y obtener sus datos de entrada, que serán nuestros requerimientos. Después, tendremos otro módulo para inferir los datos y obtener mucha más información acerca de sus preferencias. Necesitaremos otro módulo para ponderar los libros de cara a la recomendación y un último módulo mostrar por pantalla los tres ejemplares de libros recomendados.

#### 5.2.1 input

El primer subproblema que tenemos que afrontar es el hecho de preguntarle cosas al lector para poder sacar de él sus exigencias. Primeramente,

importamos todos los de templates creados desde el main, ya que almacenaremos la información en su respectivo slot cada vez que hagamos una pregunta.

En este módulo, hemos creado reglas para obtener la instancia del lector al que se hará la recomendación, su frecuencia de lectura, el lugar de lectura, el momento de lectura, y los géneros y época que exige, si es que quiere alguno en concreto. Por último, preguntaremos si de los libros que recomendamos, quiere que sean best-sellers, si los prefiere, o si le es indiferente.

### **5.2.2 inferencia-datos**

El segundo subproblema que tenemos que afrontar será inferir los datos del lector. A partir de su instancia, podremos determinar los libros que ha adquirido. De estos, sacaremos el género de cada uno y podremos ver qué géneros prefiere, si le gustan autores extranjeros, si de los libros que ha adquirido tienen buenas valoraciones y por tanto se fija en las valoraciones, si tiene algún autor preferido o incluso época.

También, podremos inferir la información de la portabilidad dependiendo del lugar donde vaya a leer, o de la complejidad, dependiendo de la edad del lector, su frecuencia de lectura, momento y lugar.

### **5.2.3 ponderar-libros**

Después de obtener toda la información del lector, sea porque nos la haya indicado o nosotros mismos la hayamos inferido, lo que haremos será crear un módulo específico para que se encargue de ponderar todos aquellos libros que cumplan con sus preferencias. En la mayoría de casos, sumaremos 10 puntos a la puntuación de un libro en caso de que sus características coincidan con las del lector. Con esto, a la hora de recomendar los libros nos será más fácil porque los tendremos ordenados de mayor similitud con el autor a menor.

### **5.2.4 recomendar-libros**

Por último, tenemos un módulo para la recomendación de dichos libros. Aún sabiendo que puntuamos los libros, el lector puede exigirnos algunos en concreto. Estos tendrán mayor prioridad ante aquellos que no requiere pero están bien valorados, ya que por mucho que un libro no sea del todo recomendable pero que justo coincida con la época exacta que nos pide, debemos recomendarlo.

En resumen, un libro bien puntuado no tiene porque ser el recomendado, ya que pueden haber otros que exijan prioridad y se recomienden antes. Al solo poder recomendar tres libros, quedarían fuera de la recomendación los demás.

### **5.3 Proceso de resolución**

Una vez planteado el problema por módulos, donde cada módulo resuelve un subproblema, debemos pensar reglas tales que nos resuelvan estos subproblemas. Además de estas, también tendremos funciones, cuya diferencia es que las funciones no tienen precondition.

#### **5.3.1 Reglas**

La primera regla de todas de nuestro programa es la de bienvenida al usuario; se hará desde el main y será la que mayor prioridad, pues queremos que se ejecute la primera.

A continuación, vienen las reglas del módulo para obtener el input del usuario. Obtendremos la instancia del lector después de mostrarle una lista de lectores, donde se verá reflejado y podrá seguir los pasos para la recomendación. Le preguntaremos cuántos libros ha leído en el último año para obtener su frecuencia de lectura anual. Seguidamente, mostraremos una lista de lugares donde podrá leer y deberá escoger uno. Análogamente para el momento, género, popularidad y época del libro que quiere leer. En cuanto al género, podrá escoger más de uno. Finalmente, tendremos una regla que nos servirá para cambiar de módulo al de inferir los datos. Por ende, tendrá salience negativo, para que aparezca la última de ese módulo.

Cambiando al módulo de inferencia de datos, tenemos una regla para inferir la portabilidad donde determinaremos que un libro debe ser portable si se lee fuera de casa o de la escuela (p.ej. en la calle o en el transporte público). Otra regla para recorrer todos los libros que ha leído el lector y ver si más de la mitad de autores que lee son extranjeros y corroborar que le gustan autores extranjeros, si todas las valoraciones de esos libros superan las 4 estrellas de valoración y por tanto observa valoraciones... o básicamente, obtener los géneros de esos libros y autores por si encontramos que algún autor o género se repite muchas veces, tenerlo en cuenta porque puede ser su favorito. Lo mismo pasaría con la época.

También, tendremos una regla para ver la especialización de un autor en base a un género; relación indicada en la ontología, pero inferida. Lo sabremos obteniendo el género de todos los libros que ha escrito ese autor y sacando el que más se repita. Por último para este módulo, inferimos la complejidad de la recomendación a partir de la edad, frecuencia, lugar y momento de lectura. Nuevamente tendremos otra última regla para cambiar de módulo al de ponderación.

Ponderamos con un punto a todos los géneros por cada una de sus apariciones en los libros que ha adquirido el lector que realiza la consulta. También ponderamos 10 puntos a cada género en los que se especialice un autor preferido del lector. Por parte de los autores, también los ponderamos con un punto por cada aparición que hagan en los libros adquiridos por el lector, y a su vez, a cada autor de una época que haya leído nuestro lector, le añadiremos otro punto. Finalmente, ponderamos todos los libros. Si son portables y el lector lo requiere, le sumamos 10 a la puntuación de ese libro. Análogamente, si su valoración supera las 4 estrellas o si prefiere best-sellers. Sumaremos 20 puntos en caso de que la complejidad del lector sea mayor o igual a la del libro, 15 en caso de que se diferencie de dos puntos de complejidad por arriba.

A continuación, sumaremos tantos puntos a ese libro como puntos tenga los géneros del libro en relación con el lector, de igual manera que lo hacemos con los autores.

Para acabar, recomendaremos los tres mejores libros de nuestra pool de libros, siempre y cuando no se nos cuele otro libro más arriba a causa de las exigencias del lector. Si por lo que sea no tenemos más libros a recomendar o la petición de recomendación es muy exigente, recomendaremos tres best-sellers o los necesarios. Acabaremos mostrando tales ejemplares por pantalla con su respectiva información, así como la información del perfil del lector con su petición. Una vez recomendados los libros, el programa habrá acabado.

### **5.3.2 Funciones**

Como hemos comentado, no solo tendremos reglas, sino que también nos ayudaremos de funciones para que sea el código más legible y también podamos reutilizar el código en varias ocasiones.

La primera de las funciones que hemos implementado ha sido para formular una pregunta de rango; la respuesta será única y el usuario deberá indicar un valor dentro de dos valores y de no ser así, la pregunta se repetirá. Crearemos

otra función con la misma temática pero para que se pueda indicar más de una respuesta. Esto por ejemplo nos será útil para requerir más de un género.

También tenemos una función para obtener la instancia del lector, que será muy similar a la pregunta de rango, pero hecha a conciencia para que la pregunta muestre mensajes de salida en referencia a las instancias de lector.

Después, dos funciones matemáticas. En estas, obtendremos el máximo valor dada una lista y el índice del máximo valor también dada una lista.

## **5.4 Desarrollo de prototipos**

Inicialmente desarrollamos una ontología que no tenía en cuenta al lector que dividía los libros en función de si eran best-sellers o si eran populares. Más tarde nos dimos cuenta de que era importante añadir el lector a nuestra ontología dada su relevancia en el problema, y decidimos considerar el número de ventas y la popularidad de un libro la misma información, dividiendo así la clase libro en las subclases bestsellers y libros poco conocidos.

Una vez teníamos la ontología bien estructurada y con los conceptos claros, pasamos toda la información a clips, donde realizamos una versión del programa muy sencilla que se basaba en recomendar tres libros aleatorios de nuestra base de datos.

Más tarde, creamos un prototipo que ya recopilaba información del usuario, y con ello hacía un filtrado básico de los libros según las exigencias de género y autores.

Para la primera entrega nuestro programa podía filtrar teniendo en cuenta a parte de todas las exigencias del lector, las preferencias de este según los libros que había adquirido en un pasado, pero aún no tenía en cuenta aspectos como la complejidad o la época de los libros. Finalmente creamos el prototipo final teniendo en cuenta todos estos aspectos y mejorando la interfaz visual al recomendar los libros.

## 6 Juegos de prueba

### 6.1 Recomendación de autores extranjeros

En este juego de prueba, disponemos de un lector de 7 años de edad. Se decanta por los libros de autores extranjeros que tienen buenas valoraciones. En el último año ha leído 6 ejemplares, y tiene previsto leer en la escuela durante la mañana. No nos ha pedido ningún género ni época en concreto. Respecto a la popularidad de los libros recomendados, exige que sean populares.

En este caso, esperaríamos que el programa nos proporcione libros de autores con una nacionalidad distinta a la del lector, de una complejidad media o baja, debido a que a pesar de su corta edad, es una persona que lee frecuentemente y en un lugar y momento cómodos; este lugar no requiere que los libros sean portables. Estos libros deberán ser best-sellers.

```
-----  
                                RECOMENDACION  
-----  
  
-> Especificaciones del lector:  
  · Nombre: Mehmet Kaya  
  · Edad: 7 años  
  · Nacionalidad: turco  
  · Libros leídos: 6 al año  
  · Lugar de lectura: Escuela  
  · Momento de lectura: Mañana  
  · Generos requiere: Sin especificar  
  · Epoca requiere: Sin especificar  
  · Best-seller: Lo quiere  
  
-> Observaciones:  
  · Portabilidad recomendacion: No  
  · Complejidad resultante: 4.5  
  · Le gustan autores extranjeros: Si  
  · Observa valoraciones: Si  
  
-----
```

Nuestro sistema ha analizado el caso particular y tiene el gusto de recomendarle:

1. Norwegian Wood - Carlos Ruiz Zafon
  - Nacionalidad autor: espanol
  - Numero de paginas: 245
  - Portabilidad: No
  - Genero: Misterio
  - Epoca: Clasica
  - Complejidad: 4.0
  - Valoracion: 3.8
  - Best-seller: Si
2. Matar a un ruisenyor - Dan Brown
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 250
  - Portabilidad: Si
  - Genero: Thriller
  - Epoca: Renacentista
  - Complejidad: 0.0
  - Valoracion: 4.6
  - Best-seller: Si
3. Cándido, o el optimismo - Jane Austen
  - Nacionalidad autor: britanico
  - Numero de paginas: 315
  - Portabilidad: No
  - Generos: Thriller, Clasica, Medieval
  - Epoca: Neoclasica
  - Complejidad: 1.5
  - Valoracion: 4.3
  - Best-seller: Si

Observamos que el sistema experto nos ha proporcionado libros que cumplen las características esperadas en base a las preferencias que hemos indicado inicialmente. Estos ejemplares son de un autor con nacionalidad diferente a la del lector, de una complejidad baja, y sin darle importancia a la portabilidad de los libros. La única anomalía que podríamos destacar, es que a pesar de la importancia que le da el lector a las buenas valoraciones, hemos obtenido un libro con una valoración por debajo de 4; esto se debe a que el programa ha considerado, debido a los géneros y autores de sus antiguas lecturas, que este libro se ajustaría más a los gustos del lector.

## 6.2 Recomendación de autores nacionales

En este juego de prueba, disponemos de un lector de 9 años de edad. Se decanta por los libros de autores nacionales que tienen buenas valoraciones. En el último año ha leído 1 ejemplar, y tiene previsto leer en el transporte público durante la noche. Nos ha pedido género policiaco, sin especificar una época en concreto, y sin darle importancia a la popularidad del libro.



```
-----
                                RECOMENDACION
-----

-> Especificaciones del lector:
  · Nombre: James Bond
  · Edad: 9 años
  · Nacionalidad: britanico
  · Libros leidos: 1 al año
  · Lugar de lectura: Transporte Publico
  · Momento de lectura: Noche
  · Generos requiere: Policiaca
  · Epoca requiere: Sin especificar
  · Best-seller: Indiferente

-> Observaciones:
  · Portabilidad recomendacion: Si
  · Complejidad resultante: 2
  · Le gustan autores extranjeros: No
  · Observa valoraciones: Si

-----
```

Nuestro sistema ha analizado el caso particular y tiene el gusto de recomendarle:

1. El cuaderno de Noah - Jules Verne
  - Nacionalidad autor: frances
  - Numero de paginas: 315
  - Portabilidad: No
  - Genero: Policiaca
  - Epoca: Moderna
  - Complejidad: 4.0
  - Valoracion: 4.0
  - Best-seller: No
2. Harry Potter y la piedra filosofal - J.R.R. Tolkien
  - Nacionalidad autor: britanico
  - Numero de paginas: 315
  - Portabilidad: Si
  - Generos: Aventuras, Romantica, Policiaca
  - Epoca: Barroca
  - Complejidad: 6.0
  - Valoracion: 4.8
  - Best-seller: Si
3. El libro de todos los generos - Raymond Chandler
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 315
  - Portabilidad: No
  - Generos: Misterio, Fantasia, Thriller, Clasica, Medieval, Aventuras, Romantica, Policiaca, Realista
  - Epoca: Vanguardista
  - Complejidad: 8.0
  - Valoracion: 1
  - Best-seller: No

Observamos que el sistema experto nos ha proporcionado libros que cumplen las características esperadas en base a las preferencias que hemos indicado inicialmente, cumpliendo la exigencia de que los géneros de los libros contienen el policíaco. Observamos anomalías de lo que podríamos esperar debido a que la complejidad del

último libro es elevada, pero esto puede ser justificado ya que al tener este libro todos los géneros que conocemos, el sistema lo ha ponderado muy bien debido a que contiene todos los géneros preferidos del lector a parte del exigido.

### 6.3 Recomendación por género requerido

En este juego de prueba, disponemos de un lector de 38 años de edad. Tiene una gran cantidad de libros adquiridos, entre los cuales se encuentran novelas de todo tipo de género y autor. En el último año ha leído 10 ejemplares, y tiene previsto leer en casa durante la tarde. Ha concretado que tiene una especial debilidad por las novelas realistas. No tiene preferencia respecto a la popularidad de los libros recomendados.

El resultado que esperamos debería contener libros con una complejidad alta, dado que se trata de un lector adulto y de lectura frecuente leyendo en condiciones muy cómodas. Además de esto, deberían aparecer novelas del género realista. La portabilidad de los libros no debería ser relevante, ya que su lugar de lectura habitual es su propia casa.

```
-----  
                                RECOMENDACION  
-----  
  
-> Especificaciones del lector:  
  · Nombre: Carl Carlson  
  · Edad: 38 años  
  · Nacionalidad: britanico  
  · Libros leidos: 10 al año  
  · Lugar de lectura: Casa  
  · Momento de lectura: Tarde  
  · Generos requiere: Realista  
  · Epoca requiere: Sin especificar  
  · Best-seller: Indiferente  
  
-> Observaciones:  
  · Portabilidad recomendacion: No  
  · Complejidad resultante: 8  
  · Le gustan autores extranjeros: Si  
  · Observa valoraciones: No  
  
-----
```

Nuestro sistema ha analizado el caso particular y tiene el gusto de recomendarle:

1. El libro de todos los generos - Raymond Chandler
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 315
  - Portabilidad: No
  - Generos: Misterio, Fantasia, Thriller, Clasica, Medieval, Aventuras, Romantica, Policiaca, Realista
  - Epoca: Vanguardista
  - Complejidad: 8.0
  - Valoracion: 1
  - Best-seller: No
2. El sabueso de los Baskerville - Jules Verne
  - Nacionalidad autor: frances
  - Numero de paginas: 315
  - Portabilidad: No
  - Generos: Romantica, Realista
  - Epoca: Moderna
  - Complejidad: 3.0
  - Valoracion: 3.5
  - Best-seller: No
3. Juego de tronos - Raymond Chandler
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 315
  - Portabilidad: Si
  - Genero: Realista
  - Epoca: Vanguardista
  - Complejidad: 9.0
  - Valoracion: 4.9
  - Best-seller: No

Observamos que el sistema experto nos ha proporcionado libros que cumplen las características esperadas en base a las preferencias que hemos indicado inicialmente. El género realismo está presente en todas las recomendaciones. No obstante, dado que el género es una exigencia y que el número de novelas realistas en el sistema es reducido, la complejidad de las recomendaciones no está tan ajustada como se podría esperar. Una última observación es que el libro que más recomienda el sistema es aquel que pertenece a todos los géneros. La razón detrás de esto es que, al tener el lector tantos libros adquiridos y de tanta variedad, el sistema ha considerado que el lector está interesado en la mayoría de los géneros y por lo tanto le ha atribuido una gran importancia a este libro.

#### 6.4 Recomendación por época requerida

En este juego de prueba, disponemos de un lector de 25 años de edad. Se decanta por los libros de autores nacionales y no observa las valoraciones. En el último año ha leído 11 ejemplares, y tiene previsto leer en casa por la tarde. No nos ha pedido ningún género en concreto, pero nos ha pedido que sean libros renacentistas. Respecto a la popularidad de los libros recomendados, prefiere que sean populares.

-----  
RECOMENDACION  
-----

-> Especificaciones del lector:

- Nombre: María Rodríguez
- Edad: 25 años
- Nacionalidad: espanol
- Libros leídos: 11 al año
- Lugar de lectura: Casa
- Momento de lectura: Tarde
- Generos requiere: Sin especificar
- Epoca requiere: Renacentista
- Best-seller: Lo prefiere

-> Observaciones:

- Portabilidad recomendacion: No
  - Complejidad resultante: 8
  - Le gustan autores extranjeros: No
  - Observa valoraciones: No
- 

Nuestro sistema ha analizado el caso particular y tiene el gusto de recomendarle:

1. Matar a un ruiseñor - Dan Brown
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 250
  - Portabilidad: Si
  - Genero: Thriller
  - Epoca: Renacentista
  - Complejidad: 0.0
  - Valoracion: 4.6
  - Best-seller: Si
2. El Hobbit - Dan Brown
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 280
  - Portabilidad: No
  - Genero: Thriller
  - Epoca: Renacentista
  - Complejidad: 3.0
  - Valoracion: 3.7
  - Best-seller: Si
3. El extranjero - Dan Brown
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 315
  - Portabilidad: No
  - Genero: Thriller
  - Epoca: Renacentista
  - Complejidad: 3.4
  - Valoracion: 4.7
  - Best-seller: No

Observamos que el sistema experto nos ha proporcionado libros que cumplen la exigencia de que la época de los autores de los libros sea renacentista, pero podemos denotar incongruencias como que a pesar de que el lector se decanta por libros nacionales, todos son extranjeros, o el hecho de que a pesar de ser un lector avanzado, todos los libros recomendados tienen una complejidad baja. Esto se debe a que no hay libros españoles de la época renacentista, y de entre los de esta época, el sistema ha elegido libros con una complejidad baja, ya que ha considerado más interesante estas opciones en base a las características de los libros adquiridos por el lector. Vemos también que de forma acorde a las preferencias del lector la mayoría de los libros recomendados son bestsellers.

### **6.5 Requerimientos muy exigentes**

En este juego de prueba, disponemos de un lector de 16 años de edad. Se decanta por los libros de autores extranjeros y tiene en cuenta las valoraciones. En el último año ha leído 3 ejemplares, y tiene previsto leer en casa durante la tarde. Nos ha pedido concretamente que sean del género realista y de época medieval. Respecto a la popularidad de los libros recomendados, exige también que sean populares.

Esperamos que el programa nos proporcione libros de autores extranjeros, de una complejidad media, ya que es adolescente, tiene baja frecuencia de lectura anual y nos avisa que leerá en casa por la tarde, lo cual se podrá concentrar con facilidad. Además, al leer en casa supondrá que el libro podría no ser portable. No obstante, estos deberán ser best-sellers. Por último, intuimos que costará obtener libros de estas características, ya que encontrar ejemplares de dicho género y época en concreto es muy exigente.

```

-----
                                RECOMENDACION
-----

-> Especificaciones del lector:
  · Nombre: Gianluigi Lentini
  · Edad: 16 años
  · Nacionalidad: italiano
  · Libros leídos: 3 al año
  · Lugar de lectura: Casa
  · Momento de lectura: Tarde
  · Generos requiere: Realista
  · Epoca requiere: Medieval
  · Best-seller: Lo quiere

-> Observaciones:
  · Portabilidad recomendacion: No
  · Complejidad resultante: 5
  · Le gustan autores extranjeros: Si
  · Observa valoraciones: Si
-----

```

**\*\* Lo sentimos, hemos conseguido encontrar 0 libros que se ajusten a tus requerimientos  
Aqui tienes 3 bestsellers que mas se ajustan a tus preferencias:**

1. Cándido, o el optimismo - Jane Austen
  - Nacionalidad autor: britanico
  - Numero de paginas: 315
  - Portabilidad: No
  - Generos: Thriller, Clasica, Medieval
  - Epoca: Neoclasica
  - Complejidad: 1.5
  - Valoracion: 4.3
  - Best-seller: Si
2. Matar a un ruisenyor - Dan Brown
  - Nacionalidad autor: estadounidense
  - Numero de paginas: 250
  - Portabilidad: Si
  - Genero: Thriller
  - Epoca: Renacentista
  - Complejidad: 0.0
  - Valoracion: 4.6
  - Best-seller: Si
3. Harry Potter y la piedra filosofal - J.R.R. Tolkien
  - Nacionalidad autor: britanico
  - Numero de paginas: 315
  - Portabilidad: Si
  - Generos: Aventuras, Romantica, Policiaca
  - Epoca: Barroca
  - Complejidad: 6.0
  - Valoracion: 4.8
  - Best-seller: Si

Podemos observar que el sistema experto ha tenido problemas proporcionando libros que cumplen las características esperadas en base a las preferencias que hemos indicado, ya que como hemos comentado anteriormente, la recomendación era muy

exigente. Aún así, el sistema nos recomienda tres best-sellers que más se ajustan a nuestras preferencias, como el autor extranjero, la complejidad de las obras en base a la complejidad del lector, y el género de los libros que ya ha leído en el pasado. El sistema sabe que el lector se fija en las valoraciones y podemos observar como los libros recomendados están bien valorados.

## 6.6 Recomendación de best-sellers para completar

En este juego de prueba, disponemos de una lectora de 54 años de edad. Se decanta por los libros de autores extranjeros y no observa las valoraciones. En el último año ha leído 2 ejemplares, y tiene previsto leer en el transporte público en la noche. Nos ha pedido concretamente que sean del género fantasía y de época clásica. Respecto a la popularidad de los libros recomendados, no tiene preferencia.

Esperamos libros de complejidad moderada, puesto que su frecuencia de lectura es baja y, tanto el lugar de lectura como el momento no son adecuados para libros complejos. Además, se esperan libros de época clásica y género fantástico, ya que así lo exige la lectora. Por último, la popularidad de los libros no debería ser relevante y, si es posible, deberán ser portables.

```
-----  
RECOMENDACION  
-----  
  
-> Especificaciones del lector:  
  · Nombre: Emily Johnson  
  · Edad: 54 años  
  · Nacionalidad: estadounidense  
  · Libros leídos: 2 al año  
  · Lugar de lectura: Transporte Publico  
  · Momento de lectura: Noche  
  · Generos requiere: Fantasia  
  · Epoca requiere: Clasica  
  · Best-seller: Indiferente  
  
-> Observaciones:  
  · Portabilidad recomendacion: Si  
  · Complejidad resultante: 5  
  · Le gustan autores extranjeros: Si  
  · Observa valoraciones: No  
  
-----
```

Nuestro sistema ha analizado el caso particular y tiene el gusto de recomendarle:

1. El Paraíso Perdido - Carlos Ruiz Zafon

- Nacionalidad autor: espanol
- Numero de paginas: 315
- Portabilidad: No
- Generos: Fantasia, Clasica
- Epoca: Clasica
- Complejidad: 2.1
- Valoracion: 2.9
- Best-seller: No

2. La Divina Comedia - Carlos Ruiz Zafon

- Nacionalidad autor: espanol
- Numero de paginas: 315
- Portabilidad: Si
- Generos: Misterio, Fantasia, Thriller
- Epoca: Clasica
- Complejidad: 9.0
- Valoracion: 4.9
- Best-seller: No

\*\* Lo sentimos, hemos conseguido encontrar 2 libros que se ajusten a tus requerimientos  
Aquí tienes el bestseller que mas se ajusta a tus preferencias:

1. Norwegian Wood - Carlos Ruiz Zafon

- Nacionalidad autor: espanol
- Numero de paginas: 245
- Portabilidad: No
- Genero: Misterio
- Epoca: Clasica
- Complejidad: 4.0
- Valoracion: 3.8
- Best-seller: Si

Podemos observar que el sistema experto ha tenido problemas proporcionando libros que cumplen las características esperadas en base a las preferencias que hemos indicado inicialmente. Esto se debe a que solo 2 libros cumplen al mismo tiempo las exigencias de época y género. Por esta razón y con el objetivo de completar la recomendación, el sistema decide añadir el bestseller que más se acerca a las preferencias de la lectora. También se puede ver que, dado el corto dominio que cumplen las exigencias de la lectora, las preferencias de portabilidad y complejidad no se han podido priorizar en algunos casos.



## 7 Planificación del proyecto

Podemos concluir que hemos resuelto el problema planteado correctamente haciendo uso de un SBC. El transcurso del proyecto a seguido la siguiente forma:

### **Primera semana**

Nos dedicamos a leer y entender el problema, plantear porque era adecuado resolverlo con un SBC, y subrayar en el texto aquellos conceptos que creímos que nos iban a ser de utilidad a la hora de realizar la ontología.

### **Segunda semana**

Empezamos a ver los videos relacionados con el uso de Protégé, y con ello hicimos un prototipo inicial de nuestra ontología, aquí nos encontramos con problemas como no saber qué conocimiento representar como clases y cual como atributos, o qué información nos sería relevante, o si deberíamos añadir el lector a la ontología.

### **Tercera semana**

Empezamos a ver los videos del temario de CLIPS. Deberíamos haber empezado a programar y resolver los primeros subproblemas, pero nos vimos en la situación de no saber si la ontología que habíamos puesto en común con los tres miembros del grupo era del todo adecuada. Gracias a las dudas resueltas por el profesor, pudimos encarrilar la ontología y decidimos modificarla en caso que fuera necesario a medida que implementamos el sistema experto. Empezamos con una implementación muy simple que obtenía los libros de un autor.

### **Cuarta semana**

Al empezar la cuarta semana vimos que deberíamos tener una extensión del prototipo inicial, cuando realmente teníamos algo muy básico y que realmente no nos serviría para la práctica. Pero, gracias a haber realizado ese paso previo, fue crucial para entender y asimilar bien los conceptos de CLIPS; ya sabíamos cómo movernos por el código y utilizar sus comandos, ¡Vamos!

Al acabar esta semana ya teníamos un prototipo inicial, sabíamos que lo íbamos a implementar por módulos pero aún no estaban acabados. Pero teníamos claro hacia dónde nos queríamos dirigir.

### **Quinta semana**

Para esta quinta semana, y aprovechando el puente de la constitución, dejamos la práctica ya casi encaminada. Acabamos nuestros módulos y reglas, y nuestro sistema experto recomendaba 3 libros, teniendo en cuenta la portabilidad del lugar de lectura del libro, los géneros y autores preferidos del autor, por si prefería autores extranjeros,

si observaba las valoraciones, si quería best-sellers o algún género en específico. Nos quedaba aún inferir por complejidad.

En caso que las especificaciones sean demasiado concretas, nuestro sistema recomendará los tres best-sellers que más se ajusten a las preferencias del lector. Suponemos que, como mínimo, habrán 3 best-sellers en nuestro sistema para recomendar.

### **Sexta semana**

En la última semana nos dedicamos a acabar la documentación, que se nos había quedado un poco atrasada con todo el tiempo que le dedicamos al aprendizaje y la realización del código CLIPS. También acabamos de pulir la inferencia de la información del programa, mejorando la información que se muestra a la hora de recomendar los libros.