

UNIVERSITAT POLITÈCNICA DE CATALUNYA

GRAU EN ENGINYERIA INFORMÀTICA

INTEL·LIGÈNCIA ARTIFICIAL

Q1 Curs 2019-2020

Sistema Basado en el Conocimiento de Recomendación de libros

Autors:

DANIEL CANO CARRASCOSA

MARTÍ VALL OLIVEIRA

RICHARD ROGSTAD WILHELM

Professor:

SERGIO ÁLVAREZ NAPAGAO



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

ÍNDICE

1. El problema
 - 1.1. Viabilidad de la solución
 - 1.2. Fuentes de conocimiento
 - 1.3. Objetivos
2. Conceptualización
 - 2.1. Conceptualización del problema
 - 2.1.1. Creación de los libros disponibles en nuestra biblioteca.
 - 2.1.2. Cálculo de la dificultad de un libro.
 - 2.1.3. Gestión de la información del usuario.
 - 2.2. Elementos del dominio
 - 2.3. División en subproblemas
 - 2.3.1. Recogida de datos del usuario.
 - 2.3.2. Análisis de las preferencias y características del usuario.
 - 2.3.3. Presentación de una solución válida.
 - 2.4. Organización
3. Formalización
 - 3.1. Desarrollo de la ontología
 - 3.1.1. Determinación del dominio y cobertura de la ontología
 - 3.1.2. Términos importantes de la ontología
 - 3.1.3. Definición de clases y jerarquía
 - 3.2. Método de resolución.
 - 3.2.1. Abstracción de los datos.
 - 3.2.2. Asociación heurística.
 - 3.2.3. Refinamiento
4. Implementación
 - 4.1. Construcción de la ontología.
 - 4.2. Módulos.
 - 4.3. Prototipos y procedimiento incremental de construcción del modelo.
5. Juegos de Prueba
6. Conclusiones

1 El problema

El objetivo de esta práctica es comprender qué metodologías de Inteligencia Artificial se deben utilizar para así poder dar a un usuario una selección de 3 libros personalizados a los gustos de este.

La cantidad de libros que se han escrito hasta la fecha actual es inconmensurable, hasta el punto en el que si un usuario quisiera saber que libro escoger en base a sus gustos y tuviera que hacerlo de forma ciega, es decir sin que nadie le guiará en sus decisiones ni en qué tipo de libros leer, la cantidad de esfuerzo por parte del usuario sería considerablemente alta, llegando incluso a dejar de leer, por haber hecho una selección de libros que no estaba alineada a sus gustos.

Para las recomendaciones de los libros nuestro sistema se basará en una colección de libros de diversos géneros y en características personales del usuario, de forma que cada usuario recibirá para él de forma personalizada una recomendación de 3 libros.

Los datos que recogeremos del usuario se pueden dividir en 3 tipos:

1. Datos referentes al género: Debido que el género es un factor muy importante a la hora de escoger un libro, ya que este influye directamente tanto en los posibles escenarios, como en los personajes o el desarrollo de la historia.
2. Datos Personales del usuario: Estos datos nos permitirán saber si un usuario debería o no leer un libro, debido a las características de este o si un libro es más apto que otro para este usuario.
3. Gustos del usuario: Estos datos nos permitirán saber que libros son potencialmente recomendables al usuario ya que este tiene gustos que coinciden con las características de los libros.

Viabilidad de la solución

Como se ha explicado previamente, este problema consiste en encontrar 3 libros entre un conjunto de N libros, basándonos en los gustos del usuario, datos personales e información de los libros de los que disponemos aparte. Por lo tanto es un problema de búsqueda, en el cual tenemos que revisar la totalidad de los libros de los que disponemos y encontrar los 3 libros que más se adecuan al usuario.

Intentar hacer este tipo de búsqueda sin un SBC es realmente costoso y complejo debido a que no disponemos de ninguna forma efectiva de navegar por todo el espacio de estados. No obstante si usamos las características de un SBC, con reglas y restricciones esta complejidad se reduce hasta algo que podemos manejar adecuadamente, por lo tanto el uso de un SBC es adecuado para este problema.

Fuentes de conocimiento

La fuentes de conocimiento es aquella información sobre la cual el SBC se basa y aprende a cómo actuar.

En nuestro caso para informarnos primeros sobre libros, su posible dificultad, así como el número de páginas hemos consultado en páginas de internet fiables, como Amazon, observando los comentarios de los compradores de los libros o en páginas como La casa del Libro entre otras.

Aunque aún teniendo toda esta información aún nos hace falta la información más importante que es la del propio usuario que nos tiene que proporcionar el mismo, ya que es en base a esta información que seremos capaces de analizar y seleccionar los 3 libros que debemos recomendarle a este.

No obstante nos encontramos un problema en lo que respecta el análisis de los libros, y es que el único atributo que tenía realmente un peso muy importante en la decisión de uno u otro libro era el género de este, ya que otros aspectos como la posible dificultad dependen mucho de usuario a usuario, ya que no es solo su experiencia lectora sino también el ambiente en el que se han criado o incluso características de su carácter más sutiles. Todo

esto hace difícil premiar un dato más que otro, por lo que suponemos que toda información que nos de el usuario es igual de importante.

Objetivos

Para que el sistema pueda cumplir con su objetivo debería ser capaz de resolver las siguientes propuestas:

- Interactuar con el usuario para obtener su información que nos servirá para elegir que tipos de libros le gustarán más, ya que estos nos proporcionará datos personales del propio usuario y datos más relacionados con gustos de libros o experiencia de lectura para decidir de forma más acorde al usuario.
- Aprovechar al máximo la base de datos que hemos creado y toda la información que se puede abstraer a partir de esta
- Entregar al usuario de forma elegante y ordenada el resultado obtenido, indicando la preferencia de qué libros se adecuan más a sus gustos.

2 Conceptualización

Después de haber identificado el problema que estamos tratando de resolver, comenzamos la fase de conceptualización donde el problema se desglosó en subproblemas. Las consideraciones más importantes a tener en cuenta al sugerir un libro en nuestra opinión es el gusto del usuario por la temática del libro, ya que otros elementos como la dificultad serán igualmente muy importantes, pero la temática del libro influirá en tanto a la historia, personajes o escenas de acción del libro mismo. Será desmotivador leer un libro que esté por encima del nivel del lector y será aburrido si es demasiado fácil. Cada libro también tiene muchas características que a un usuario le pueden gustar o no, por ejemplo, si el libro tiene una longitud muy extensa, ya que quizás el usuario prefiere lecturas breves.

2.1 Conceptualización del problema

Creación de los libros disponibles en nuestra biblioteca

Como en la mayoría de las bibliotecas, hemos dividido los libros en géneros y subgéneros. Como el enunciado de práctica nos restringe a incluir solo el género de fantasía, hemos dividido los libros en los siguientes subgéneros:

- Adventure
- Epic
- Magic
- Sword and Sorcery
- Fabel
- Superheroes
- Cyperpunk
- Science Fiction
- Romantic
- Dark
- Spooky

Para simplificar el problema, solo hemos asignado un subgénero a un libro, sin embargo, hemos hecho cuatro grupos que consten en géneros con similitudes. Nuestra biblioteca

consta de 52 libros, con al menos tres en cada género, este último hecho es debido a que para esta práctica, debemos recomendar siempre 3 libros, por lo tanto necesitamos como mínimo 3 en cada uno de los géneros para asegurar que existe una solución.

Cálculo de la dificultad de un libro

Hay principalmente dos factores que determinan el nivel de dificultad en un libro. En primer lugar, es la dificultad del lenguaje que se encuentra en el libro. ya que si el libro consta de muchas palabras desconocidas o si las descripciones se vuelven demasiado excesivas y es difícil seguir la historia. En segundo lugar, en general, la extensión del libro también puede indicar la dificultad, ya que requiere más dedicación y concentración para terminar un libro largo que uno corto.

Gestión de la información del usuario

El sistema obtendrá información sobre el usuario a través de una interacción en la que el usuario debe responder a varias preguntas del sistema. Las respuestas se utilizarán para hacer un perfil del usuario. Las preguntas son las siguientes:

1. *What type of setting do you prefer on books, medieval/fantasy or modern/urban?*
2. *What do you prefer in a story, a lot of action or more character and world development?*

Las preguntas 1 y 2 se utilizarán para determinar qué tipo de género podría ser adecuado para el usuario. Hay que tener en cuenta que estas preguntas pueden ser más fáciles de responder que “¿qué género prefieres?”, ya que el usuario puede tener inconsistencia o ninguna experiencia sobre géneros. Además, una pregunta tan directa también podría filtrar demasiado la solución, lo que no es bueno.

3. *How old are you?*

La pregunta 3 se utilizará para determinar el nivel de dificultad de los libros que sean adecuados. Aunque existen excepciones, es probable que los jóvenes hayan leído menos, mientras que las personas mayores tienen más experiencia.

También es más razón para asumir esto, ya que vivimos en una generación digital donde los niños prefieren ver series en el ordenador que leer libros.

4. *How many hours do you read per week?*

La pregunta 4 se utilizará de muchas maneras, porque dice algo sobre el estilo de vida del usuario y de qué manera la lectura debe incorporarse en la vida del usuario. En primer lugar, dice algo sobre la dificultad adecuada y la extensión del libro, ya que esto es una indicación de la dedicación necesaria para terminar el libro. En segundo lugar, también podemos interpretar qué tipo de popularidad prefiere el usuario. Por ejemplo, si no tiene mucho tiempo para leer durante una semana, deberíamos recomendar un libro más vendido o popular, ya que es más probable que esto haga que el lector siga interesado en el libro. Por otro lado, un usuario que pasa mucho tiempo leyendo libros puede tener un interés especial en la literatura, por lo que las críticas pueden ser más adecuadas para el usuario.

5. *About how many pages have the books you usually read?*

La pregunta 8 dirá algo sobre la experiencia del lector, así ayudará al sistema a determinar el nivel de dificultad del libro.

6. *What Language do you speak?*

7. *Do you speak any other languages?*

Las preguntas 6 y 7 son triviales, ya que es esencial encontrar libros escritos en un idioma que sea comprensible para el usuario. En esta práctica asumimos que la nacionalidad del autor es el idioma del libro.

8. *How many books have you read?*

La pregunta 8 sirve para determinar la experiencia del lector. La edad no es el único factor a tener en cuenta al elegir un nivel de dificultad adecuado, ya que pueden existir niños que han leído mucho y personas mayores que nunca han tenido tiempo de leer hasta ahora. Por lo tanto, las preguntas 3, 4, 5 y 8 combinadas determinarán el nivel de dificultad.

9. *Do you like any of these writers?*

10. *Do you like any other writers?*

11. *Do you have any genre preferences?*

12. *Do you have any other genre preferences?*

Al final de la filtración, el sistema podría tener que priorizar entre una selección de libros, por lo que debe hacer preguntas más directas. Las preguntas 9 a 12 preguntarán sobre autores y géneros concretos, y con esta información será posible elegir tres y solo tres libros.

2.2 Elementos de dominio

Autor:

- Nombre
- Apellido
- Libro(s)
- Nacionalidad
- Dificultad (si sus libros consisten en muchas palabras difíciles y son excesivamente descriptivos)

Libro:

- Título
- Autor
- Género
- Dato de publicación
- Numero de paginas
- Popularidad
- Lenguaje (la dificultad de la lenguaje en el libro, es decir lo pesado es que leer)

2.3 División en subproblemas

Recogida de datos del usuario

Al iniciar el programa, el usuario tiene que contestar una serie de preguntas sobre sus características (edad, cuantos libros leído y tiempo que dedica para leer a la semana, etc.). El segundo paso es elegir preferencias sobre los libros de qué disponemos en nuestra biblioteca. Seguidamente se plantean unas preguntas más directas para hacer la última filtración. Todo este proceso facilita al experto formar una solución personalizada.

Análisis de las preferencias y características del usuario

El experto se dedica a analizar las respuestas ofrecidas por el usuario. La forma en que se ha planteado este subproblema es con un campo *puntuación*. Según las preferencias y características del usuario se le atribuyen más puntos a unos libros que a otros.

Presentación de una solución válida

Después de que el sistema haya verificado con éxito la información del usuario con las características de los libros, el sistema necesita encontrar los tres libros que se corresponden mejor con el perfil del usuario, es decir los libros que tengan más puntajes. A partir de entonces, debe presentar la solución ordenada de la mejor coincidencia a la menor coincidencia de esos tres libros.

2.4 Organización

Como mencionamos antes, sabemos que este es un problema que SBC puede resolver. Con la información que obtenemos del usuario, crearemos la solución a través de los siguientes pasos:

- Recopilar datos concretos del usuario.
- Abstracción de los datos.
- Obtención de los datos abstractos.
- La asociación heurística.
- Presentación de la solución

3 Formalización

3.1 Desarrollo de la Ontología

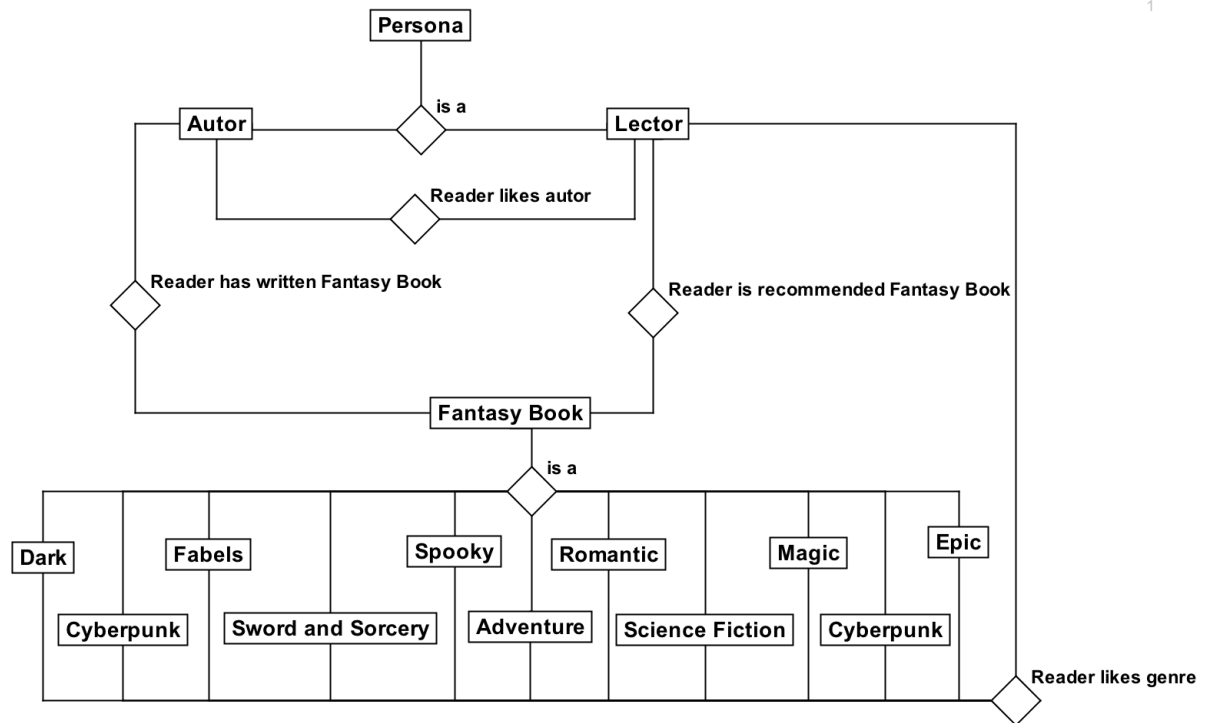
Abstracción de los datos

En el apartado de conceptualización hemos descrito ya todos los conceptos relevantes al dominio del problema, por lo que serán estos los usados para la ontología en este problema. La ontología que hemos desarrollado ha de ser capaz tanto de clasificar los libros en diferentes grupos y conservar sus características como a la vez contener todos los Autores de estos libros para así poder acceder a su información, ya que hay ocasiones en las que tendremos necesidad de acceder a estos autores para obtener información de estos.

Términos importantes de la ontología

Los términos más importantes de la ontología son los **libros**, puesto que al fin y al cabo, se trata de un sistema de recomendación de libros y por lo tanto la solución propuesta va ser un conjunto de libros, y los **autores**, ya que estos son la referencia más grande que tenemos de los libros para poder decidir si son adecuados para el lector fuera de las características básicas de estos, como las páginas o el género.

Definición de clases y jerarquía



La clase lector no es una clase implementada como tal, sino que está implementada de forma abstracta en forma de de templates en el código, ya que la la clase persona no deja de ser una clase usada para guardar información del usuario.

Autor

Autor		
Name	String	
Surname	String	
Libro	Instances*	Libro_Fantasia
Nationality	Symbol	
Difficulty	String	

Define el autor de un libro, la clase se relaciona con los libros escritos por el autor.

Los atributos son:

- **Name:** Nombre del autor.
- **Surname:** Apellido del autor.
- **Libro:** Los libros escritos por el autor.
- **Nationality:** La nacionalidad del autor.
- **Difficulty:** La dificultad media de las obras del autor.

Libro Fantasía

Libro Fantasía		
Title	String	
Pages	Integer	
Popularity	Symbol	
Year	Integer	
Language	Symbol	
Writer	Instance*	Writer

Esta clase contiene toda la información del libro necesaria, la cual consta de el título de este, necesario pues a la hora de recomendar al usuario un libro deberemos escoger tres y mostrarselos al usuario. Igualmente es importante el numero de paginas, popularidad, año, dificultad del lenguaje y escritor, ya que todos estos nos proporcionarán la información necesaria para decidir si este es adecuado o no al usuario.

- **Title:** Indica el título de la obra
- **Pages:** Numero de paginas del libro
- **Popularity:** Como de popular es el libro
- **Year:** Año en el que el libro fue publicado
- **Language:** Dificultad en el lenguaje del libro
- **Writer:** Escritor del libro

3.2 Método de resolución

Abstracción de los datos

Para hacer la abstracción, primeramente, el sistema hace preguntas al lector para obtener algunos datos específicos referentes a su experiencia como lector, como el tipo de narrativa que más le gusta, el número de libros que ha leído o las horas que dedica a la lectura semanalmente. A partir de aquí abstraemos de estos datos:

- Un grupo de géneros: Abstraídos a partir de las preguntas de ambiente de la narrativa y estilo de la historia.
- Una dificultad de lectura: Abstraída a partir de la edad, las horas de lectura semanales y los libros leídos.
- Una popularidad específica: Abstraída a partir de las horas de lectura y los libros leídos
- Unas extensiones preferidas: Una de las cuales abstraída del número de páginas y la otra abstraída de las horas de lectura semanales y los libros leídos.

Las normas para generalizar los datos del lector son las siguientes:

1. Para el concepto de la edad, la generalizamos de la siguiente forma:

AGE	
Entre 5 y 17	YOUNG
Entre 18 y 30	ADULT
Igual o mayor que 31	OLD

2. Para el concepto de número de horas semanales de lectura, la generalizamos de la siguiente forma:

Nº HOURS	
Entre 0 y 2	OCCASIONALLY

Entre 3 y 7	CASUAL
Entre 8 y 14	REGULAR
Igual o mayor que 15	DEDICATED

3. Para el concepto de numero de paginas hacemos la siguiente generalización de información:

Nº PAGES	
Entre 0 y 150	SHORT
Entre 151 y 400	AVERAGE
Igual o mayor que 401	LONG

4. Para el concepto de idioma que habla el usuario hacemos la siguiente generalización.

NATIONALITY	
Spain	Spanish
United Kingdom or United States	English
Norwegian	Norway
French	France
Japanese	Japan
Grece	Greek

5. Para el concepto del número de libros leídos, hacemos la siguiente generalización:

Nº BOOKS	
Entre 0 y 5	BEGINNER
Entre 6 y 15	AMATEUR

Igual o mayor que 16	ADVANCED
----------------------	----------

Las normas para la abstracción de datos son las siguientes:

→ Para la dificultad:

◆ EASY DIFFICULTY SI:

→ SI YOUNG AND OCCASIONALLY OR CASUAL

→ SI ADULT AND OCCASIONALLY

◆ MEDIUM DIFFICULTY SI:

→ SI YOUNG AND AMATEUR

→ SI YOUNG AND BEGINNER AND REGULAR OR DEDICATED

→ SI YOUNG AND ADVANCED AND OCCASIONALLY OR CASUAL OR REGULAR

→ SI ADULT AND BEGINNER AND CASUAL OR REGULAR OR DEDICATED

→ SI ADULT AND AMATEUR OR ADVANCED AND OCCASIONALLY

→ SI OLD ADULT AND OCCASIONALLY OR CASUAL AND BEGINNER OR AMATEUR

◆ HARD DIFFICULTY SI:

→ SI YOUNG AND DEDICATED AND ADVANCED

→ SI ADULT AND CASUAL OR REGULAR OR DEDICATED AND AMATEUR OR ADVANCED

→ SI OLD ADULT AND (REGULAR OR DEDICATED) OR (OCCASIONALLY OR CASUAL AND ADVANCED)

→ Para los géneros:

◆ NARRATIVE ACTION AND SETTING MEDIEVAL

→ ADVENTURE, EPIC, MAGIC, SWORD_AND_SORCERY

◆ NARRATIVE DEVELOPMENT AND SETTING MEDIEVAL

→ FABULA, ADVENTURE, MAGIC, DARK

◆ NARRATIVE ACTION AND SETTING URBAN

→ SUPERHEROES, CYBERPUNK, SCI-FI

◆ NARRATIVE DEVELOPMENT AND SETTING URBAN

→ ROMANTIC, DARK, SPOOKY

→ Para la popularidad:

◆ BEST SELLER

→ BEGINNER AND CASUAL OR OCCASIONALLY

◆ POPULAR:

→ BEGINNER AND REGULAR

→ AMATEUR AND CASUAL OR OCCASIONALLY

◆ CRITIC

→ AMATEUR AND REGULAR

◆ BEGINNER AND DEDICATED

→ ADVANCED AND CASUAL OR OCCASIONALLY

◆ NORMAL

→ ADVANCED AND REGULAR

→ AMATEUR AND DEDICATED

◆ NON-POPULAR

→ ADVANCED AND DEDICATED

→ Para la extensión:

◆ SHORT EXTENSION SI:

→ SI AMATEUR AND OCCASIONALLY

→ SI BEGINNER AND OCCASIONALLY OR CASUAL OR REGULAR

◆ NORMAL EXTENSION SI:

→ SI BEGINNER AND DEDICATED

→ SI AMATEUR AND CASUAL OR REGULAR

→ SI ADVANCED AND OCCASIONALLY

◆ LONG EXTENSION SI:

→ SI AMATEUR AND DEDICATED

→ SI ADVANCED AND CASUAL OR REGULAR OR DEDICATED

Asociación heurística

Se relacionan los datos abstractos obtenidos con una de las cuatro soluciones abstractas. Concretamente, a partir de los géneros escogidos sacamos un conjunto de libros (que son los que corresponden a esos géneros) y puntuamos positivamente a los libros de ese conjunto que tengan la misma dificultad, popularidad o extensión que hemos abstraído de los datos, de manera que le sumaremos un punto al puntaje del libro por cada uno de estos valores que coincida con los valores abstraídos del lector (Ej. Si hemos abstraído dificultad difícil, se puntuará positivamente los libros del conjunto de solución abstracta que se correspondan a esa dificultad).

Refinamiento

Finalmente, una vez tenemos una solución abstracta debemos convertirla en una solución real, que en nuestro caso significa escoger 3 libros de esa solución abstracta. Para ello, miramos los libros que tienen más puntaje (es decir que más coinciden con los datos abstractos de extensión, dificultad y popularidad) y de estos libros intentamos coger 3. Si hay varios libros empatados, entonces el sistema hará un par de preguntas sobre géneros preferidos y autores preferidos al usuario y acabará de determinar los mejores libros entre los que están empatados. Para que no vuelva a darse un empate, en esta ocasión utilizamos un puntaje de tipo float (con decimales) y aparte de las dos nuevas preguntas

también sumamos al puntaje del libro un valor decimal entre 0 y 1 que representa lo mucho que se acerca el número de páginas del libro al número de páginas que nos ha contestado el usuario anteriormente (Ej. Si el usuario nos ha dicho que lee libros de 432 páginas puntuaremos con +1 los libros que tengan exactamente 432 páginas, y cuanto más se separe el número de páginas de un libro de este valor más se acercará la puntuación que le daremos a 0). De este modo conseguimos que sea casi imposible que los libros tengan un mismo puntaje porque es muy difícil que coincidan exactamente en el número de páginas. Así pues, para obtener la solución final solo tenemos que coger los 3 libros de mayor puntuación y los mostramos por pantalla con orden de preferencia de más recomendado a menos.

4 Implementación

4.1 Construcción de la Ontología y de templates utilizados

Hemos generado la ontología con Protégé, con las características descritas en el correspondiente apartado. A parte hemos creado diversos de templates para poder manejar la información de forma más cómoda utilizando diversas reglas. El nombre e información de dichos templates es la siguiente :

- **npages:** Usado para guardar la cantidad de páginas que suelen tener los que lee el usuario, y que se utiliza para priorizar unos libros por delante de otros en caso de que haya más de 3 libros potenciales a ser recomendados.
- **solutions_abs:** Contiene los géneros de aquellos libros que formarán la solución abstracta, existiendo 4 posibles grupos, los cuales son:
 1. Aventura , Épica, Magia, Espada y Hechicería.
 2. Fábula, Aventura, Magia, Terror.
 3. Superhéroes, Cyberpunk, Ciencia Ficción, Épica
 4. Tematica Oscura o tenebrosa, Romántica, Terror, Cyberpunk

- **likes:** Aquí nos guardamos la mayoría de datos del usuario, tales como los lenguajes que habla, si le gustan libros cortos, más extensos, si prefiere categorías Best Seller o no, etc, etc.
- **books:** Aquí nos guardamos los títulos de los libros asociados a un puntaje, el cual determinará cuales de estos N libros serán aquellos tres que finalmente recomendamos al usuario, y donde la puntuación de un libro será mayor o menor dependiendo de cómo se ajuste el libro a los gustos de este último.
- **lista:** aquí guardamos todos los deftemplates de books con sus respectivas puntuaciones, pero en esta lista los libros se ordenan de mayor a menor puntuación, para así facilitar su manipulación a la hora de presentar al usuario los libros que más se adecuan a su selección.
- **filtro:** En este deftemplate se guardan los libros, que después de haber hecho las suficientes preguntas como para identificar qué libros son potenciales a recomendar, aun nos quedan diversos libros por discutir, por ejemplo en el caso de tener exactamente 2 libros que se ajusten a los deseos del autor, pero tener otros 100 que a primera vista parecen todos iguales si solo nos basamos en las anteriores preguntas, entonces estos 100 se incluirían en este deftemplate y se harán pregunta muy específicas, únicamente sobre estos últimos libros para ver qué libro de entre estos 100 se adecua más al usuario.
- **solution:** Si en el deftemplate filtro hay 100 libros potencialmente iguales a recomendar, aquí se encuentran los otros libros que ya sabemos que vamos a recomendar, este deftemplate se usa principalmente para conservar esta información.
- **autoresf:** Aquí se guardan los autores que el usuario dice que le gustan, este template solo se usa en caso de tener que hacer un desempate final en cuanto a la selección de libros si vemos que demasiados cumplen con los gustos de este, haciéndolos a ojos del sistema indistinguibles en cuanto a importancia.
- **generof:** Aquí preguntamos directamente, dentro de los géneros que hemos obtenido a partir de la solución abstracta, cuales de ellos le gusta más al usuario, al

igual que `autoresf` es una pregunta muy específica que tiene como objetivo, eliminar las últimas posibles discrepancias que puedan haber a la hora de seleccionar los últimos 3 libros.

- **autores:** Aquí nos guardamos los autores de aquellos libros que a ojos del sistema son indistinguibles en cuanto a prioridad, este `deftemplate` principalmente sirve para facilitarnos la manipulación de estos
- **stop_gen:** Este `deftemplate` lo usamos para hacer más fácil el uso de una regla a la hora de comprobar los géneros que le gustan al usuario
- **stop_aut:** Este `deftemplate` lo usamos para hacer más fácil el uso de una regla a la hora de comprobar los autores que le gustan al usuario
- **stop_pag:** Este `deftemplate` lo usamos para hacer más fácil el uso de una regla a la hora de comprobar los géneros que le gustan al usuario

A parte de todos estos `deftemplates` para soportar nuestra implementación, hemos incluido instancias de la clase `Autores` con slots que nos permiten extraer información como su nombre, su nacionalidad y los libros que ha escrito y también tenemos instancias de `Libros` que contienen información como la fecha de publicación, número de páginas y otras que nos permite abstraer información para así decidir si ese libro es o no apto para el usuario., para así tener una base de datos. Todos los `deftemplates` descritos previamente se instancian de forma dinámica.

4.2 Módulos

Por supuesto nuestro equipo era consciente de la existencia de los módulos y cómo estos podrían sernos de utilidad a la hora de implementar todo nuestro modelo de SBC, pero por falta de tiempo y debido a que finalmente logramos implementar todas las características que inicialmente nos habíamos propuesto sin mucha dificultad, decidimos que no era necesario utilizar ningún módulo, excepto el que ya tenemos por defecto, el módulo `MAIN`, donde se encuentran todas las reglas y funciones necesarias para llevar a cabo la ejecución de este programa.

4.3 Prototipos y procedimiento incremental de construcción del modelo

Para desarrollar nuestro SBC, hemos seguido una serie de pasos para cada vez ir incrementando más y más nuestro modelo. En primera instancia decidimos hacer un conjunto de reglas simples que hicieran diversas preguntas al usuario, y guardar estas preguntas en un `deftemplate`, por supuesto para hacer esto tuvimos que hacer primero una única `defrule` inicial que comenzará todo este proceso. Pero para hacer estas preguntas básicas necesitábamos de algún método para preguntar y obtener la información, por lo que al final tuvimos que hacer 3 funciones que se basaban en o bien hacer preguntas de si o no, en hacer preguntas más complejas como escoger una opción dentro de una lista de posibilidades o bien de hacer una pregunta donde se podía escoger un número pero este tenía que estar acotado a un cierto rango de valores.

Una vez ya teníamos las preguntas básicas hechas, procedimos a hacer una abstracción de la información a partir de estas, como ya hemos dicho previamente utilizamos el hecho de que tipo de temática le gusta al autor y de ahí abstraemos unos géneros. Toda el resto de información que preguntamos la conservamos para así poderla utilizar para, dentro de este conjunto de libros que forman parte de la solución abstracta, decidir cuales son los 3 libros que más le gustarán a nuestro usuario.

Pero para hacer todo esto tuvimos que hacer un más `defrules`, a si que una vez que ya tuvimos las preguntas funcionales, a parte con la abstracción de nuestras preguntas para decidir la solución abstracta, implementamos una `defrule` que hacía la asociación de la abstracción con la solución abstracta pertinente, la cual llamemos `abstract_solution`.

Una vez que tuvimos hasta este punto tuvimos que hacer diversas `defrules` para así poder puntuar y diferenciar, de entre los libros que teníamos en nuestra solución abstracta cuales eran mejor y cuales peores entre todos estos. (Fue hasta este punto donde entregamos en la entrega de nuestro prototipo para el punto extra de la Assignatura).

A partir de este punto ya podríamos haber dejado como estaba nuestro modelo, pero nos encontramos en la situación de que en muchos casos nuestras preguntas dejaban un grupo

de 3 o 4 libros donde las puntuaciones de estos libros eran equivalentes, así que decidimos que en caso de darse esta situación hacer una serie de preguntas mucho más concretas restringidas únicamente a estos libros, para así disipar las posibles discrepancias restantes, pero para esto tuvimos que hacer una defrule donde se encargará de hacer todo este análisis de si se daba esta situación, y en caso de ser así (ya que era una situación común pero no siempre se daba), entonces se procedía a hacer las otras preguntas sobre únicamente estos libros, para dar así finalmente la solución final.

5 Juegos de prueba

Para poder comprobar la efectividad y calidad de nuestro sistema y testear los diferentes casos posibles hemos diseñado los siguientes seis juegos de pruebas. A continuación siguen unas instancias actuales, los casos que hemos planteado, la salida esperada, para los mismo y la solución que el sistema ha ofrecido:

INSTANCIAS ACTUALES

Para los dos siguientes casos de prueba estamos en una situación donde las instancias son todos aquellos libros que o bien son en español, inglés o japonés. Incluyendo únicamente libros de superhéroes, épicas y oscuro.

Caso inventado (1)

Una persona de 14 años que no tiene mucho tiempo para leer. Ha leído pocos libros hasta ahora, pero dentro de la media, y los libros han sido de extensión media. Le gustan libros de medieval y acción, además habla español y inglés. No le gusta ningún autor mejor que otro, y no hay nada preferencias de género.

Solución esperada

Dado que el usuario es joven, pero parece tener algo de experiencia, el sistema enfrentará el dilema de elegir el nivel de dificultad. Aún así esperamos que tenga más en cuenta la experiencia, por lo que esperamos libros de alta dificultad. Dado que no ha dedicado mucho tiempo a la lectura, esperamos que el sistema sugiera clásicos o best sellers, de autores ingleses o españoles. Además, el género debería ser épica.

Output

System of Recommendation for Books

What type of setting do you prefer on books, medieval/fantasy or modern/urban? (Medieval/Urban)

Medieval

What do you prefer in a story, a lot of action or more character and world development? (Action/Development)

Action

How old are you? (INTEGER NUMBER)

14

How many hours do you read per week? (INTEGER NUMBER)

2

About how many pages have the books you usually read? (INTEGER NUMBER)

250

What Language do you speak?

Spanish

Norwegian

English

French

Japanese

Greek

Spanish

Do you speak any other languages?

Norwegian

English

French

Japanese

Greek

None

English

Do you speak any other languages?

Norwegian

French

Japanese

Greek

None

None

How many books have you read? (INTEGER NUMBER)

3

Do you like any of these writers? (Write only the surname)

Patrick Rothfuss

John Ronald Reuel Tolkien

Brandon Sanderson

George Raymond Richard Martin

None

None

Do you have any genre preferences?

None

Epic

None

We recomend you the following list of books

1. The Two Towers

2. The Return of the King

3. Mistborn: The Final Empire

CLIPS>

Caso inventado (2)

Una persona de 26 años que no ha leído nada antes. Como necesita un nuevo pasatiempo, quiere leer un libro para saber si le gusta leer o no. Basado en los series que le gustan, prefiere los libros urbanos con desarrollo. Además, tiene mucho tiempo para leer y solo habla inglés. No conoce ningún autor, pero prefiere libros de género épica.

Solución esperada

El sistema volverá a tener un dilema para determinar el nivel de dificultad, ya que la persona se considera adulta pero no tiene experiencia. Esta vez, esperamos que se propongan libros de media dificultad. Como el lector tiene mucho tiempo pero carece de experiencia, esperamos los libros críticos de escritores ingleses. También deberían ser libros de género dark. A pesar de cree que le gustan los libros épicos, sus preselecciones no lo indican.

Output

```
System of Recommendation for Books

What type of setting do you prefer on books, medieval/fantasy or modern/urban? (Medieval/Urban)
Urban
What do you prefer in a story, a lot of action or more character and world development? (Action/Development)
Development
How old are you? (INTEGER NUMBER)
26
How many hours do you read per week? (INTEGER NUMBER)
15
About how many pages have the books you usually read? (INTEGER NUMBER)
0
What Language do you speak?
Spanish
Norwegian
English
French
Japanese
Greek

English
Do you speak any other languages?
Spanish
Norwegian
French
Japanese
Greek
None

None
How many books have you read? (INTEGER NUMBER)
0
Do you like any of these writers? (Write only the surname)

Osamu Dazai
Agatha Christie
Sabina Murray
None

None
Do you have any genre preferences?
None
Dark

None
We recomend you the following list of books
1. Indigno de ser humano
2. A carnivores inquiry
3. The seven deadly Sins
CLIPS>
```

INSTANCIAS ACTUALES

Para los dos siguientes casos de prueba estamos en una situación donde las instancias son todos aquellos libros que han sido publicado después de 1950. Incluyendo únicamente libros de fábulas, románticos o ciencia ficción.

Caso inventado (3)

Una persona de 62 años que ha trabajado como profesor de lectura y obviamente ha leído muchísimo en su vida. Además, los libros que ha leído han sido en su mayoría largos. Le gustan libros de desarrollo y urbano, y tiene mucho tiempo para leer. El habla español, inglés, francés y noruego. Le encanta el autor Dave Duncan y prefiere libros románticos.

Solución esperada

Ahora esperamos libros con gran dificultad y muchas páginas. Como tiene mucha experiencia y mucho tiempo para leer, esperamos que el sistema recomiende libros no populares. No esperamos libros de autores griegos. Al menos uno de los libros debería ser de autor Dave Duncan o ser de género romántico.

Output

*En la siguiente página por falta de espacio

```
System of Recommendation for Books

What type of setting do you prefer on books, medieval/fantasy or modern/urban? (Medieval/Urban)
Urban
What do you prefer in a story, a lot of action or more character and world development? (Action/Development)
Development
How old are you? (INTEGER NUMBER)
62
How many hours do you read per week? (INTEGER NUMBER)
30
About how many pages have the books you usually read? (INTEGER NUMBER)
800
What Language do you speak?
Spanish
Norwegian
English
French
Japanese
Greek

Spanish
Do you speak any other languages?
Norwegian
English
French
Japanese
Greek
None

English
Do you speak any other languages?
Norwegian
French
Japanese
Greek
None

French
Do you speak any other languages?
Norwegian
Japanese
Greek
None

Norwegian
Do you speak any other languages?
Japanese
Greek
None

None
How many books have you read? (INTEGER NUMBER)
30
Do you like any of these writers? (Write only the surname)

Robert James Waller
Dave Duncan
None

Duncan
Do you like any other writers? (Write only the surname)

Robert James Waller
None

None
Do you have any genre preferences?
None
Romantic

Romantic
Do you have any other genre preferences?
None

None
We recomend you the following list of books
1. Orgullo y prejuicio
2. Toda la verdad de mis mentiras
3. Magic Casement
CLIPS>
```

Caso inventado (4)

Una madre quiere utilizar el sistema para encontrar un libro ella puede leer para su hijo de 5 años cada noche. El libro hay que ser fácil, por eso ella pone no experiencia y que ha leído libros de pocas páginas antes. También piensa que su hijo prefiere acción y medieval, y ella le gustan autores japoneses por sus cómicas.

Solución esperada

Esperamos libros de corta extensión y baja dificultad. Además esperamos libros de autores japoneses porque ella pone japonés como idioma y le gusta *Otomo*. Los libros deberían ser best sellers por la baja dificultad y relativamente poco tiempo para leer durante las semanas.

Output

```
System of Recommendation for Books
```

```
What type of setting do you prefer on books, medieval/fantasy or modern/urban? (Medieval/Urban)
```

```
Medieval
```

```
What do you prefer in a story, a lot of action or more character and world development? (Action/Development)
```

```
Action
```

```
Sorry but we don't have any existence of the book that you're looking for
```

```
CLIPS>
```

Comentario de output: Obtenemos este resultado porque las preselecciones *medieval* y *action* indican géneros que no existen en las instancias actuales.

INSTANCIAS ACTUALES

Para los dos siguientes casos de prueba estamos en una situación donde las instancias son todos aquellos libros que tienen más que 250 páginas. Incluyendo únicamente libros de mágicos, cyberpunk y de miedo.

Caso inventado (5)

Una persona de 51 años no ha leído bastante mucho durante su vida. Durante los fines de semana ella le encanta sentarse por unas horas para leer un libro. Ella le gustan libros urbanos con desarrollo, y normalmente han sido largos. Ella habla griego y español.

Solución esperada

Como ella es adulta y antes ha leído libros largos esperamos libros de gran dificultad, aunque no ha leído mucho. Parece que ella tiene un buen momento para leer, por eso esperamos libros normales de autores griegos y españoles. Además los libros deberían ser de miedo.

Output

```
System of Recomendation for Books
What type of setting do you prefer on books, medieval/fantasy or modern/urban? (Medieval/Urban)
Urban
What do you prefer in a story, a lot of action or more character and world development? (Action/Development)
Development
How old are you? (INTEGER NUMBER)
51
How many hours do you read per week? (INTEGER NUMBER)
5
About how many pages have the books you usually read? (INTEGER NUMBER)
500
What Language do you speak?
Spanish
Norwegian
English
French
Japanese
Greek

Greek
Do you speak any other languages?
Spanish
Norwegian
English
French
Japanese
None

Spanish
Do you speak any other languages?
Norwegian
English
French
Japanese
None

None
How many books have you read? (INTEGER NUMBER)
2
Do you like any of these writers? (Write only the surname)
George Orwell
Rosa Montero
Stephen King
None

None
Do you have any genre preferences?
None
Cyberpunk
Spooky

None
We recomend you the following list of books
1. 1984
2. Lágrimas en la lluvia
3. The Dark Tower
CLIPS> █
```

Caso inventado (6)

Una persona de 37 años le gusta mucho leer, pero su vida es muy ocupada, entonces no tiene mucho tiempo para leer. Ella ha leído mucho como niño, pero los últimos años casi nada. Le gustan libros de medieval y desarrollo, especialmente el género mágico, y normalmente los libros son de media extensión. Además le gusta el autor J. K. Rowling. Aparta habla inglés, Español y Japonés.

Solución esperada

Obtenemos algunas contradicciones en términos de dificultad, ya que el es adulto y relativamente experimentado, pero solo ha leído libros mediano. Además, los libros del género esperan magia porque prefiere tanto el género mágico como el autor J. K. Rowling. Como tiene poco tiempo para leer, esperamos libros críticos o populares.

Output

```
System of Recomendation for Books
What type of setting do you prefer on books, medieval/fantasy or modern/urban? (Medieval/Urban)
Medieval
What do you prefer in a story, a lot of action or more character and world development? (Action/Development)
Development
How old are you? (INTEGER NUMBER)
37
How many hours do you read per week? (INTEGER NUMBER)
3
About how many pages have the books you usually read? (INTEGER NUMBER)
400
What Language do you speak?
Spanish
Norwegian
English
French
Japanese
Greek

English
Do you speak any other languages?
Spanish
Norwegian
French
Japanese
Greek
None

Spanish
Do you speak any other languages?
Norwegian
French
Japanese
Greek
None

Japanese
Do you speak any other languages?
Norwegian
French
Greek
None

None
How many books have you read? (INTEGER NUMBER)
6
We recomend you the following list of books
1. Circus
2. The Outsider
3. Bag of Bones
CLIPS> █
```

Comentario de salida: Como el sistema ya ha encontrado exactamente 3 libros, el usuario no recibe las preguntas directas sobre qué autor y género prefiere.

6 Conclusiones

Hemos conseguido implementar un sistema experto de recomendación de libros que cumpla con los requisitos establecidos por la práctica usando la metodología de SBC mediante análisis heurístico. Sin embargo, hay mucho margen para mejorar. Hay muchas ideas que hemos tenido pero que ya sea por falta de tiempo o de recursos no hemos podido implementar. Algunos ejemplos serían:

- Utilizar una puntuación variable personalizado en vez de una puntuación constante, de manera que dependiendo del usuario puntúe más unos campos que otros. Por ejemplo, que para un usuario, que el libro sea del género adecuado puntúe 2 en vez de uno, y que para otro usuario puntúen 2 la extensión o la dificultad. De este modo, la recomendación sería mucho más personalizada y ajustada para cada usuario.
- Añadir más preguntas sobre otros aspectos que puedan influir en el tipo de libros recomendados, como por ejemplo los sitios donde lee habitualmente el usuario (cosa que por ejemplo podría influir en recomendar libros más pequeños o fáciles de transportar si se lee fuera de casa o en libros más robustos si solo se lee en casa).
- Añadir preguntas sobre la personalidad y mentalidad del usuario, de manera que se pueda tomar en cuenta también si un libro es adecuado para un usuario acorde a su sensibilidad y madurez intelectual como lector.
- Utilizar una base de datos de amigos/conocidos ampliable que nos permita relacionar varios usuarios, y en función de como pueda afectarles la opinión de los demás, utilizar los datos de sus amigos para ajustar más la recomendación de los libros.
- Ampliar la gama de libros, pues al fin y al cabo, al existir tantos libros, es prácticamente imposible tenerlos todos en una única ontología, por lo que siempre hay margen a mejorar en este aspecto.