



Universitat Politècnica de Catalunya

FACULTAT D'INFORMÀTICA DE BARCELONA

RECOMENDACIÓN DE LIBROS

Sistemes Basats en el Coneixement

Aubach Altès, Artur
Fuster Palà, Llum
Gálvez Serrano, Paula
Molina Sedano, Òscar

Índex

1	Introducció	1
2	Identificació del problema	2
2.1	Descripció	2
2.2	Viabilitat del SBC	2
2.3	Objectius	3
3	Anàlisi de les fonts de coneixement	4
3.1	Fonts de coneixement factual	4
3.2	Fonts de coneixement del domini	5
3.2.1	Entrevista amb un Expert	5
3.2.2	Altres	6
4	Conceptualització	8
4.1	Conceptes del domini	8
4.2	Resolució Teòrica	8
4.3	Problemes i Subproblemes	9
4.3.1	Entendre les Preferències del Lector	9
4.3.2	Recomanar un Llibre amb la Informació Obtinguda	10
5	Formalització	11
5.1	Mètode de Creació de l'Ontologia	11
5.2	Fases de Desenvolupament de l'Ontologia	11
5.3	Mètode de Resolució	16
6	Creació de la Base de Dades Sintètica	18
6.1	Generador de Dades	18
6.2	Conversió de les Instàncies	23
7	Implementació	24
7.1	Representació de l'Ontologia	24
7.2	Mòduls de Resolució	28
7.2.1	Mòdul MAIN	28
7.2.2	Mòdul INPUT	29
7.2.3	Mòdul BASE	29
7.2.4	Mòdul BABY	29
7.2.5	Mòdul CHILDREN	29
7.2.6	Mòdul TEEN	30
7.2.7	Mòdul ADULT	30
7.2.8	Mòdul TRIGGER_W	30
7.2.9	Mòdul ACT	30
7.2.10	Mòdul SOLUTION	30
7.2.11	Mòdul PUNCTUATION	31
7.2.12	Mòdul JUSTIFY	31
7.3	Mètode de Resolució	31
7.4	Prototips i Mètode Incremental	31
7.4.1	Prototip 1 - Solució Bàsica	32
7.4.2	Prototip 2 - Poda per Edat	33
7.4.3	Prototip 3 - Regles Específiques	35
7.4.4	Prototip 4 - Regles Específiques (Part 2)	38
7.4.5	Prototip 5 - Question skip, Sistema de Puntuació i Poda per TriggerWarnings	41
7.4.6	Prototip Final	42
8	Jocs de prova	46
8.1	Jocs de prova a partir d'instàncies	46

8.2	Jocs de prova amb executador	47
8.2.1	Generació de Comandaments	47
8.2.2	Execució Automàtica de Jocs de Prova	47
8.2.3	Exemple d'ús	48
8.2.4	Jocs de prova manuals	48
8.3	Comparació dels dos mètodes	48
9	Futures Ampliacions	49
9.1	Atribut dificultat	49
9.1.1	Habit de lectura-aduts	49
9.2	Atribut Sagas	49
9.3	Atribut Idiomes	49
9.4	Mètode de puntuació més complex	50
9.5	Mètode de justificació més complex	50
9.6	Historial de llibres	50
10	Conclusions	51
11	Appèndix 1	52

1 Introducció

En l'era actual de l'abundància d'oferta de literatura i la creixent presència de llibreries en línia, la elecció d'una lectura adequada pot resultar aclaparadora. Per abordar aquest desafiament, s'han desenvolupat sistemes de recomanació de llibres que busquen proporcionar als lectors suggeriments personalitzats basats en una varietat de factors. No obstant això, la majoria d'aquests sistemes sovint es queden curts quant a la precisió i la profunditat de les seves recomanacions. Sovint, la recomanació proporcionada per un sistema en línia manca de la calidesa i la intuïció que obtindríem d'un llibreter de confiança o un amic apassionat per la lectura.

En aquest context, el nostre treball es centra en el domini de les novel·les de ficció, explorant la creació d'un sistema de recomanació de llibres altament personalitzat i efectiu. El nostre objectiu és emular la forma en què un expert en recomanació de llibres consideraria diverses dimensions per comprendre el perfil del lector i les seves preferències literàries.

Ens submergirem en la complexa tasca d'entendre els lectors, prenent en compte factors que van més enllà de la simple elecció de gèneres literaris, com la inclinació cap a llibres populars, les valoracions crítiques o la dificultat del llibre. A més, analitzarem la importància dels gèneres literaris i els autors preferits, que poden proporcionar pistes valuoses sobre les preferències d'un lector.

Aquest enfocament ens permetrà proporcionar recomanacions de llibres altament personalitzades, considerant la individualitat de cada lector. La nostra documentació es basarà en investigacions sòlides i fonts de qualitat per comprendre els elements clau de la recomanació de llibres, des de gèneres literaris fins a autors destacats.

2 Identificació del problema

2.1 Descripció

El problema consisteix en realitzar un recomanador de llibres a partir d'un sistema basat en el coneixement. És una tasca complexa degut al gran nivell de personalització que requereix el fet d'adaptar-se a les preferències concretes del lector, a més dins un món on l'oferta literària és molt àmplia. L'espai de cerca es refereix a la totalitat de l'oferta literària que s'hauria de considerar. En realitat, aquest espai de cerca s'aproxima a la grandària d'un catàleg d'una llibreria. Haurem d'intentar representar al màxim aquest catàleg per assegurar-nos que les recomanacions són adequades i variades.

Hi ha diversos elements principals que són clau per a la implementació del sistema: el lector, els llibres i els autors.

El lector, amb les seves preferències, hàbits de lectura i context específic, és el nucli al voltant del qual gira la qüestió. Els llibres, amb les seves múltiples dimensions com ara gènere, autors, temes i estils, són el camp de selecció. Els autors, amb el gènere principal que escriu, el públic a qui es dirigeix i altres factors, també són rellevants per a la relació d'aquest amb els llibres.

Les interrelacions entre aquests elements són crucials. Un enteniment profund del perfil del lector, que incorpora factors com l'edat, l'atractiu cap als bestsellers o la confiança en les crítiques literàries, entre d'altres, ha de ser relacionat amb els atributs diversificats dels llibres disponibles i els autors. Aquesta relació no és trivial i requereix una comprensió detallada tant dels lectors com dels llibres i els seus autors.

En la implementació, haurem de considerar una metodologia que permeti una captura detallada i dinàmica del perfil del lector i que estigui preparada per navegar a través de la diversa varietat de llibres disponibles, identificant aquells s'adaptin millor amb les inclinacions úniques del lector. La personalització, la precisió i la capacitat d'explicar les recomanacions de manera transparent són consideracions clau que guiaran la nostra aproximació.

El nostre repte, per tant, és crear un sistema que integri aquests elements diversos, abordi les seves complexitats i proporcioni recomanacions que no només compleixin les preferències explícites dels lectors, sinó que també revelin i satisfacin les seves inclinacions implícites. Això millorarà significativament l'experiència de descobrir i gaudir de la literatura en tota la seva riquesa i diversitat.

2.2 Viabilitat del SBC

En aprofundir en la complexitat del problema, identifiquem que la natura multifacètica de la recomanació de llibres exigeix un enfocament que pot gestionar dades heterogènies i oferir sortides personalitzades basades en una àmplia gamma de criteris. Les característiques dels lectors, com ara les seves preferències, hàbits i restriccions de temps, juntament amb les diverses dimensions que defineixen un llibre, com ara el gènere, l'autor, l'estil i les valoracions, contribueixen a la complexitat del domini.

La capacitat dels Sistemes Basats en el Coneixement (SBC) per manejar la **complexitat**, la incertesa i l'ambigüitat en la presa de decisions els fa ser una opció viable per aquest escenari. Un SBC basat en regles és especialment apte, donat que permet la creació, manipulació i aplicació de regles que poden ser tan simples o complexes com sigui necessari per adaptar-se a les diverses capes de complexitat dins del domini de recomanació de llibres.

Una de les raons pel qual creiem fermament en la viabilitat del SBC en aquest context és la seva flexibilitat inherent. Un sistema basat en regles pot ser modificat i ampliat amb relativa facilitat, permetent la incorporació de nous llibres, autors, gèneres i altres criteris pertinents en el sistema sense necessitat de reprogramar o revisar l'estructura bàsica del sistema.

Això no només facilita l'actualització i l'ampliació del sistema a mesura que emergeixen noves tendències literàries i preferències dels lectors, sinó que també suporta la capacitat del sistema per oferir recomanacions altament **personalitzades**. La natura dinàmica del SBC permet la integració fluida de les dades del perfil del lector, les valoracions dels llibres, les ressenyes i altres dades pertinents per formar un quadre comprensiu dels interessos i necessitats del lector.

La **interpretabilitat i explicabilitat** és una altra força clau del SBC. Les decisions i recomanacions generades per un sistema basat en regles poden ser explicades i justificades de manera transparent, oferint als usuaris una comprensió clara de com i per què es van formar determinades recomanacions. Aquesta transparència no només augmenta la confiança dels usuaris en el sistema, sinó que també proporciona oportunitats per a la retroalimentació i l'ajustament, facilitant una millora contínua i l'adaptació a les necessitats dels usuaris en evolució.

En resum, creiem que un SBC basat en regles no només és viable, sinó també altament efectiu per abordar els desafiaments associats amb la recomanació de llibres en un món caracteritzat per una diversitat i complexitat significatives. La seva flexibilitat, adaptabilitat, escalabilitat i transparència són atributs essencials que, creiem, no només abordaran les insuficiències dels sistemes de recomanació existents, sinó que també elevaran l'experiència de recomanació a un nivell de personalització i precisió anteriorment inassolible.

2.3 Objectius

L'objectiu principal d'aquest projecte és desenvolupar un Sistema Basat en el Coneixement (SBC) per a la **recomanació de llibres** que sigui capaç d'oferir fins a tres suggeriments de llibres altament personalitzats per a l'usuari. Aquest SBC haurà de tenir en compte una àmplia gamma de variables relacionades tant amb les característiques específiques dels llibres com amb les preferències personals i els hàbits de lectura dels usuaris. A més, el sistema haurà d'explicar de manera clara i concisa les raons per les quals cada llibre ha estat recomanat, augmentant així la transparència i la confiança dels usuaris en les recomanacions proporcionades.

L'objectiu d'aquest treball no és únicament obtenir un sistema de recomanació. La importància no cau solament sobre el desenvolupament, sinó que el procediment complet és molt rellevant. Així doncs, podem definir uns objectius secundaris que s'han assolit durant el desenvolupament del treball.

Aprendre a Utilitzar Protégé: Assimilar les habilitats i el coneixement necessaris per utilitzar eficientment Protégé, una eina essencial per a la creació, edició i manipulació d'ontologies, facilitant així la representació estructurada i detallada del coneixement dins del domini de recomanació de llibres.

Dominar CLIPS: Adquirir una comprensió profunda i pràctica de CLIPS, un llenguatge potent per a la programació de sistemes basats en el coneixement, per garantir que el nostre SBC sigui eficient, escalable i adaptable als canvis i a les necessitats emergents.

Us Solvent dels Recursos d'Informació: Desenvolupar la competència en l'ús efectiu i ètic de diverses fonts d'informació, garantint que el coneixement integrat en el SBC sigui no només ampli i profund, sinó també fiable, autèntic i respectuós amb els drets de propietat intel·lectual.

Implementació Metodològica: Aplicar una metodologia rigorosa en l'anàlisi, disseny, implementació i avaluació del SBC, assegurant que cada etapa del desenvolupament sigui ben planificada, executada i documentada.

Jocs de Prova Representatius: Seleccionar i utilitzar jocs de prova representatius per validar la funcionalitat, la fiabilitat i la precisió del SBC, i oferir una explicació detallada dels resultats obtinguts per demostrar la seva eficàcia i precisió.

Aquests objectius secundaris estan alineats amb la meta global d'assegurar que el **SBC** desenvolupat no només sigui tècnicament sòlid i funcional, sinó també èticament responsable, usuari-centrat, i adaptable a un entorn en constant evolució. Aquest enquadrament garantirà que les recomanacions de llibres proporcionades pel sistema siguin personalitzades, precises, transparents i valuoses per als usuaris, contribuint significativament a enriquir la seva experiència de lectura.

3 Anàlisi de les fonts de coneixement

Per construir un sistema de recomanació de llibres, hem vist adient constar d'un conjunt ampli d'instàncies a recomanar. Seguint amb l'estructura que hem extret de l'anàlisi del problema, necessitem 2 tipus principals d'instàncies (llibres i autors). A més a més, l'usuari final estarà present com a instància única de lector. Per tant, necessitem d'un conjunt de llibres i els seus respectius autors amb la seva informació.

El primer recurs d'obtenció d'instàncies de llibres i autors que ens vam plantejar va ser de dades reals. Per dur-ho a terme, havíem de trobar una font fiable d'informació que permetés l'ús i extracció de les seves dades i a més a més que s'apropessin al format o estructura que desitjàvem per a la construcció de l'ontologia.

3.1 Fonts de coneixement factual

El procés de cerca d'una font fiable d'informació factual ens ha suposat tot un repte. Tant que, com explicarem en aquest apartat, hem decidit afrontar el problema de l'obtenció de dades de manera sintètica creada per nosaltres mitjançant un programari en Python.

A l'enunciat de la pràctica se'ns proposaven un parell de recursos web: [OpenLibrary](#) i [ISBNDB](#). A continuació analitzarem l'estructura, funcionalitat i ètica d'ús de cadascuna de les fonts provades durant el període de selecció.

OpenLibrary OpenLibrary és una llibreria pública que, a més a més, proporciona una API per a poder obtenir informació de pràcticament qualsevol llibre donat el seu identificador ISBN. A banda de l'API, OpenLibrary deixa a la disposició de tothom els anomenats *dumps*, que són fitxers `.txt` que contenen totes les dades en format `.json`.

Però això suposa també certs problemes. Per començar, si volguéssim optar per l'API, hauríem de tenir un llistat d'ISBNs per a extraure la seva informació. Sinó, podríem fer un *sampling* de les bases de dades completes i, amb coneixements d'SQL obtenir el nombre que volguéssim d'instàncies de llibres ben relacionades amb els seus autors. Tot i així es presenta el següent problema: valors nuls. Hi ha molts llibres que tenen camps buits, i per tant hauríem de dissenyar una estratègia per a tractar-los o directament eliminar-los. Per acabar, el problema principal resideix en l'absència d'algun dels camps que ens agradaria tenir, com per exemple *rating*.

ISBNdb Respecte a la informació factual que podríem extreure de ISBNdb, a part de què torna a requerir d'una llista d'ISBNs, tampoc té algun dels camps que ens permetrien recomanar de millor manera un llibre als usuaris. Un exemple de camp que falta és el del pes del llibre.

GoodReads Durant el procés d'obtenció de dades, vam optar per agafar un fitxer `.csv` que contenia informació de 52.000 llibres extreta de la web de [GoodReads](#) mitjançant *web scrapping*. La base de dades era molt completa i constava de pràcticament totes les característiques que buscàvem. Tot i això, però, vam adonar-nos que l'extracció de dades havia incomplert els termes d'ús de la font original, cosa que careix d'ètica, per la qual cosa també vam haver de renunciar a usar-la en el nostre treball.

De fet, la manera d'adquirir la informació de totes les entrades web i datasets publicats a internet que hem visitat era qüestionable. L'exemple de la pàgina web de GoodReads sol ser el recurs per excel·lència de la majoria d'elles i en totes elles s'incompleix un dels principals punts dels termes d'ús i servei de la font.

6. Not sublicense or redistribute Goodreads data to any 3rd parties.

Aquest punt consisteix en la impossibilitat de redistribuir informació extreta de la font a tercers. Per a més informació consulta la següent [pàgina web](#).

Ara bé, aquesta font tenia a disposició una API mitjançant la qual et podies registrar i obtenir informació de manera gratuïta i legal. De ser així, agafar una base de dades d'algué que hagués seguit aquest procediment simplement hagués comportat un estalvi de temps i esforç de part nostra (a banda

d'un incompliment de la seva part del punt 6 dels termes d'ús). Però, èticament parlant, mentre ens registressim, no estariem aprofitant-nos de la font sense el seu consentiment.

Desgraciadament, GoodReads va deixar de donar claus de desenvolupador gratuïtes per accedir a la seva API al desembre de l'any 2020. Així que actualment tenen privatitzades les seves dades i, qualsevol ús que en fem d'elles podria considerar-se que anem en contra de la seva voluntat i els seus drets de propietat.

3.2 Fonts de coneixement del domini

3.2.1 Entrevista amb un Expert

Per a crear la base de dades amb una certa **coherència i justificació**, es necessita **coneixement expert** que recolzi les decisions que preses sobre quines són les característiques més importants a tenir en compte o els percentatges a triar. Per aquest motiu, i per tal d'assegurar que el nostre SBC sigui tant tècnicament robust com rellevant per a les necessitats de l'usuari, s'ha optat per fer una entrevista amb un expert en el sector.

A més a més, vam intentar cercar aquesta informació via internet, però la majoria de pàgines no tenien una entitat oficial que respallés el seu contingut. Tot i així algunes ens van ser útils per conduir les decisions preses.

A continuació, es presenta el disseny, desenvolupament i anàlisi d'aquesta entrevista, que ens servirà per a tenir una base sòlida sobre la qual construir les dades del nostre projecte. Després veurem alguns recursos web d'on n'hem pogut treure profit.

Investigació prèvia

Abans de la entrevista, es va dur a terme una investigació exhaustiva per identificar possibles candidats amb un perfil i experiència que poguessin aportar valor al nostre projecte sobre la recomanació de llibres. Després d'avaluar diverses opcions, el nostre equip va optar per la Laura, que té una combinació única d'experiència pràctica i coneixement en l'àmbit que ens interessa.

Perfil de la Laura

- **Edat:** 28 anys
- **Experiència Professional:** Dependenta a "La Casa del Llibre" durant 1 any.
- **Relevància per al Projecte:** L'experiència de la Laura treballant directament amb clients en la recerca de llibres la converteix en una font inestimable d'informació. Ha interactuat amb diversos perfils de clients i té una visió pràctica de com les persones trien llibres en funció dels seus gustos, necessitats i contextos.

Eixos temàtics a tractar

- **Criteris de Recomanació:** Busquem entendre els mètodes i lògiques que utilitza la Laura per recomanar un llibre a un client. Això ens permetrà programar algorismes de recomanació més precisos.
- **Característiques dels Llibres i la seva Relació per a la Creació d'una Base de Dades:** Pretenem comprendre quines característiques dels llibres són crucials per classificar-los adequadament en una base de dades.
- **Relació de Gèneres i Subgèneres:** Volem explorar com es relacionen diferents gèneres i subgèneres de llibres per construir una taxonomia que pugui ser implementada en el SBC.

Conclusions de l'entrevista

A partir de la informació de la entrevista hem encaminat la definició de les classes i les preguntes a fer. Tot i així, per facilitar la creació de la base de dades hem fet algunes simplificacions, com limitar un autor a un sol gènere i a un sol públic).

Es rellevant la pregunta obligatòria sobre l'edat del client, la qual cosa ens proporciona una guia clara per a la classificació inicial i la filtració de instàncies del nostre SBC. El gènere també va ser una característica destacada com a essencial per a una recomanació d'un llibre. No obstant això, es va fer evident que no és essencial preguntar explícitament pel "gènere que es busca" en certs contextos, com quan es tracta de nens de 4-6 anys, on les preferències estan menys definides i les recomanacions poden ser més generals.

La Laura va destacar la importància dels 'trigger warnings' en recomanar un llibre, especialment per a adolescents i adults. Això destaca la necessitat d'integrar un sistema d'alertes o etiquetes en el nostre SBC que informi els usuaris sobre possibles continguts sensibles o gràfics en els llibres recomanats, assegurant així recomanacions apropiades i segures.

Amb tot això, ampliem amb detalls específics per a cada secció mencionada, incloent exemples pràctics basats en la informació adquirida de l'entrevista amb la Laura:

Selecció de Preguntes L'entrevista amb la Laura va revelar la necessitat d'introduir preguntes més específiques i adaptatives. Per exemple, una pregunta com "Quins 'triggers' t'agradaria evitar?" ens permet ajustar les recomanacions basades en les sensibilitats personals dels lectors. A més, preguntes sobre les preferències de format, com "Prefereixes llibres en tapa dura o tapa blanda?", optimitzen les recomanacions segons les preferències físiques del llibre dels lectors.

Filtratge per Edat La informació aportada per la Laura ha estat clau per implementar un filtratge més afinat basat en l'edat. Per exemple, per a un nen, el sistema podria introduir preguntes com "Quines són les teves aficions?", mentre que per als adults, podria enfocar-se en preferències literàries més complexes, amb preguntes com "Tens una preferència per les novel·les modernes o clàssiques?".

Gènere i Subgènere Gràcies a ella, hem guanyat *insights* valuoses sobre la complexitat de la relació entre gèneres i subgèneres. Ara, si un usuari està interessat en la ciència ficció, el nostre SBC podria aprofundir amb opcions com "Cyberpunk" o "Dystopia Postapocalíptica", oferint una experiència més personalitzada i adaptada.

La integració d'aquests elements detallats i específics, gràcies a la informació recollida de l'entrevista amb la Laura, ha potenciat la capacitat del nostre SBC de proporcionar recomanacions que són a la vegada tècnicament acurades i altament personalitzades. Cada pregunta, criteri de filtratge i la comprensió ampliada dels gèneres i subgèneres ens permeten ara atendre de manera més efectiva les diverses necessitats, gustos i preferències dels lectors.

El desenvolupament complet de l'entrevista es troba a l'Apèndix 1

3.2.2 Altres

Hem necessitat més fonts de coneixement del domini per tal de crear una base de dades sintètica.

Web1 <https://universoabierto.org/tag/estadisticas/> és un Blog de la biblioteca de Traducción y Documentación de la Universidad de Salamanca. D'aquí obtenim informació sobre el percentatge de llibres venuts segons el gènere, i la edat dels lectors.

Web2 [AQUESTA](#) és una pàgina web de la *Comunidad Baratz*, un espai de difusió i comunicació de notícies i informació relacionades amb el món de les biblioteques, els llibres o la lectura.

Aquest article analitza els resultats de les preguntes relacionades amb els hàbits de lectura, extretes de les enquestes realitzades pel Baròmetre del CIS (Centre d'Investigacions Sociològiques). A més, també incorpora un estudi realitzat per l'AMIC (Associació per la Investigació de Mitjans de Comunicació) que es centra en els hàbits de lectura de les persones que utilitzen Internet com a mitjà de consum d'informació i cultura.

Mitjançant aquesta pàgina web obtenim els percentatges dels llibres llegits segons el gènere i l'edat.

Web3 En la fase de generació de noms per a la nostra base de dades, hem optat per utilitzar la tecnologia de generació de text proporcionada per ChatGPT, específicament la versió "gpt-3.5-turbo-

16k-0613". Aquesta eina ens ha permès crear noms únics i creatius que poden servir com a títols per a llibres, noms d'autors, entre d'altres, en la nostra base de dades sintètica.

Per tal de garantir l'originalitat i evitar l'ús de noms amb llicència o protegits, vam especificar en el prompt que la generació de noms havia de ser creativa, i no una extracció de noms existents d'internet. Un exemple de prompt utilitzat per a aquesta tasca va ser el següent:

CONTEXT:

Estoy en búsqueda de títulos únicos para libros, los cuales deben encapsular la esencia del contenido de manera artística. Cada título debe ser un reflejo de la profundidad y riqueza de la historia contenida en sus páginas. Busco originalidad y creatividad, evitando la repetición y similitud tanto en estructura como en estilo. Cada título debe ser una pieza única que invite a la curiosidad y al asombro.

QUERY:

Genérame una lista de {cantidad} títulos para libros del género {genero} dentro del subgénero {subgenero}. Cada título debe ser un mundo en sí mismo, evitando similitudes y repetición en estructura, tono o estilo. Considera la diversidad en la elección de palabras, la estructura gramatical y la evocación emocional.

OUTPUT:

Quiero una lista de títulos en formato de array: ["libro1", "libro2", ...]

En resposta, la API de ChatGPT generava una llista de títols en el format desitjat.

És important assenyalar que, tot i haver especificat la necessitat de crear noms en lloc d'extreure'ls, hi ha una possibilitat remota que el sistema pugui generar per casualitat noms que coincideixin amb títols o noms reals amb llicència. Segons les [polítiques d'ús](#) de OpenAI, es fomenta l'ús segur i responsable de les seves eines, i per això, han establert polítiques específiques per a tots els usuaris de les seves models, eines i serveis. La responsabilitat de la generació de noms que puguin infringir drets de tercers recau en nosaltres com a usuaris de la tecnologia. Entenem i assumim les conseqüències de l'ús de la tecnologia i la gestió dels noms generats.

En cap moment aquest treball pretén tractar aquests llibres i autors com a reals. Volem deixar explícitament clar que totes les dades usades són sintètiques i no pretenen copiar ni plagiar dades de llibres.

Web 4 Per la generació dels noms dels autors hem utilitzat una API open-source "<https://randomuser.me/api/>".

4 Conceptualització

4.1 Conceptes del domini

El domini del nostre sistema són els llibres de ficció. Segons l'anàlisi anterior del problema hem fixat un conjunt de propietats que ens agradaria saber d'un llibre i els seus autors. Si fóssim capaç d'obtenir-les, constaríem d'unes dades molt complertes amb les quals podríem desenvolupar totes les idees i regles de producció que tenim en ment per tal de recomanar un llibre de la forma més sòlida possible.

Les característiques principals que voldríem saber d'un **llibre** són:

- Títol
- Autor
- Número de pàgines
- Pes
- Llengua original
- Gènere(s) literari(s)
- Preu
- *Rating*
- ...

Per altra banda, les característiques principals d'un **autor** que voldríem saber són:

- Nom
- Edat (Data de naixement)
- Gènere literari especialitzat
- Llengua original
- ...

Aquesta informació en concret correspon al coneixement factual del domini, que s'anirà modificant a mesura que obtinguem les dades i anem implementant el sistema.

Adicionalment, cada llibre ha d'estar vinculat al seu autor de manera que, si obtenim informació sobre les característiques de l'autor, puguem identificar tots els llibres dels autors que comparteixin aquestes característiques. A més, el gènere literari especialitzat de l'autor haurà de ser el mateix que el gènere dels seus llibres.

Finalment, cal tenir en compte que, per aconseguir determinades característiques dels llibres que el lector desitja, no sempre formularem les preguntes directament. En molts casos, deduirem quin tipus de llibres satisfarà les preferències del lector a partir de les seves respostes. Per exemple, no preguntarem al lector quant vol que pesi un llibre, sinó que preguntarem el tipus de tapa i el nombre de pàgines. A partir d'aquí es podrà deduir el pes del llibre.

4.2 Resolució Teòrica

Com a metodologia de disseny, hem pres la decisió d'aplicar la metodologia de software àgil en comptes de seguir una aproximació en cascada tradicional. Aquesta metodologia implica realitzar una primera fase de cascada inicial ràpida, la qual no requereix una preparació tan exhaustiva. A partir d'aquest punt, es va actualitzant i millorant el sistema de manera constant.

Cal tenir en compte que en les etapes inicials d'un projecte de SBC, sovint resulta complex obtenir un disseny del problema que sigui totalment adequat. És a dir, és probable que les decisions preses inicialment siguin inadequades i, com a conseqüència, pot comportar una redefinició radical del projecte que forci a tornar a començar durant el procés de desenvolupament. La metodologia que nosaltres hem

triat, al tenir un prototipat inicial senzill, ofereix l'avantatge de facilitar aquesta redefinició amb un menor cost i una major flexibilitat.

A més, cal tenir en compte que en els SBS és complicat fer una estimació precisa de la naturalesa i la quantitat del coneixement que haurà de ser incorporat. Per això, és millor idea anar avançant pas a pas que no tractar d'abordar tot el projecte de cop i, posteriorment, descobrir que la magnitud del coneixement requerit supera les expectatives, fent que no es pugui acabar la pràctica a temps.

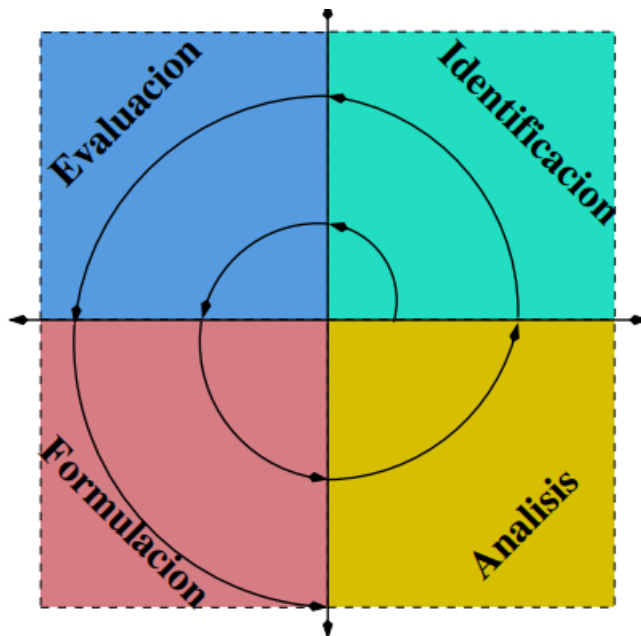


Figure 1: Estructura de la metodologia agile software

Tal i com es mostra en la figura, aquesta metodologia és un disseny continu, on a partir d'un prototipat inicial es va passant per les fases de identificació, avaluació, formulació i anàlisi trobant noves millores a cada pas i augmentant la complexitat del nostre sistema. Les diverses rodes que hem anat seguint es poden veure reflexades en els diferents prototips funcionals que es mostren en l'apartat 7.4.

4.3 Problemes i Subproblemes

Des d'una perspectiva resolutiva, el nostre problema implica diverses tasques d'anàlisi, ja que gran part de la seva complexitat es troba en la identificació de les característiques necessàries per fer recomanacions precises als lectors. No obstant això, també s'inclouen tasques de síntesi que són crucials per a l'èxit del sistema. Aquestes tasques de síntesi inclouen la eliminació de llibres que no compleixen les característiques desitjades pel lector, la creació del sistema de puntuació basat en les respostes del lector i la determinació dels llibres més adients per a cada lector.

Malgrat que el sistema implica tasques d'ambdós tipus, com és comú en problemes complexos, la seva naturalesa es decanta cap a les tasques de síntesi.

No obstant això, podem desglossar el problema principal en dos grans àmbits: primer, entendre les preferències del lector, i segon, utilitzar aquesta informació per a fer recomanacions de llibres adequats. Aquests dos àmbits principals contenen, a la seva vegada, subproblemes més específics, on cada subproblema pot ser un tipus de tasca diferent.

4.3.1 Entendre les Preferències del Lector

Aquesta tasca no és trivial. Per entendre realment què busca un lector, han sorgit diversos subproblemes que cal abordar:

- **Preguntes Adaptatives:** No podem formular les mateixes preguntes a tots els lectors. Cal determinar quines preguntes són pertinents i com adaptar-les segons el perfil del lector.
- **Classificació dels Lectors:** Si decidim adaptar les preguntes, caldria preguntar-nos si existeix una manera de classificar els lectors en diferents grups o "clusters". Si és així, quina seria aquesta classificació?
- **Experiència d'Usuari:** Per obtenir respostes sinceres i detallades, cal que el lector se senti còmode durant tota l'experiència de recomanació. Com podem optimitzar aquesta experiència?

En resum, aquest eix del problema consisteix en triar preguntes adequades per a que l'usuari tingui una bona experiència i poder classificar els lectors en certs grups. Totes aquestes tasques es consideren d'anàlisi ja que no estan generant una solució, sinó identificant els components i característiques necessàries per a poder generar la informació més endavant.

4.3.2 Recomanar un Llibre amb la Informació Obtinguda

Un cop hem entès les preferències del lector, arribem al segon gran eix del problema: fer la recomanació. Això també comporta certs subproblemes:

- **Tractament de la Informació:** Amb totes les dades recopilades, com les processem per extreure'n conclusions útils?

Les tasques a fer en aquest eix, en canvi, són de síntesi ja que has de construir un sistema de solucions a partir de les característiques triades pel lector.

Com es pot apreciar, cada problema i subproblema aporta una capa addicional de complexitat al SBC. No obstant, a mesura que anem desenvolupant prototips del SBC, s'intentarà abordar i solucionar cadascun d'aquests reptes, sempre amb l'objectiu de millorar l'experiència final del lector.

5 Formalització

5.1 Mètode de Creació de l'Ontologia

El procés de creació de l'ontologia per al nostre SBC de recomanació de llibres ha estat una tasca crucial per garantir una representació semàntica i estructurada de les dades amb les quals treballem.

Per a modelar l'ontologia hem utilitzat l'eina Protégé ja que ofereix una interfície gràfica que facilita la representació i la gestió dels conceptes, propietats i les seves relacions. Encara que al principi va ser un repte familiaritzar-se amb l'entorn de Protégé i les seves funcionalitats, amb el temps i la pràctica, hem aconseguit dominar l'eina i crear una ontologia robusta i ben estructurada. Ens ha ajudat en la tasca d'aprenentatge la [documentació](#) de Protégé

Aquesta eina ens permet extreure arxius tipus turtle (.ttl) i amb la llibreria *owl2else* hem pogut obtenir un fitxer programat en python per a la seva traducció a clips: *owl2clips*. Aquesta llibreria també ens ha proporcionat la capacitat d'extreure una representació gràfica de la nostra ontologia amb el fitxer owl2plot.

5.2 Fases de Desenvolupament de l'Ontologia

En aquesta secció, es descriuen les diferents fases involucrades en el desenvolupament de l'ontologia per a la nostra recomanació de llibres. Un dels reptes i competències més significatius d'aquesta pràctica és la capacitat de construir una ontologia des de zero. Aquesta tasca, encara que amb molts desafiaments, ens permet adquirir una comprensió profunda del domini d'interès i assegurar que l'ontologia sigui específicament adaptada a les necessitats del nostre sistema.

El procés es divideix en diverses fases per garantir que l'ontologia sigui completa, precisa i útil per a l'aplicació final. Les fases descrites inclouen:

1. Determinar el domini i cobertura de l'ontologia.
2. Considerar la reutilització d'ontologies existents.
3. Enumerar els termes importants a l'ontologia.
4. Definir les classes i la seva jerarquia.
5. Especificar les propietats de les classes.
6. Detallar les característiques de les propietats.
7. Creació d'instàncies.

En les següents subseccions, cada fase es descriurà amb més detall, incloent les decisions preses, les consideracions i els problemes afrontats. A través d'aquest procés, esperem no només desenvolupar una eina útil per a la recomanació de llibres, sinó també demostrar la nostra capacitat de construir i utilitzar ontologies d'una manera efectiva i significativa.

Fase 1: Determinar el domini i cobertura de l'ontologia

La determinació del domini i la cobertura de l'ontologia és el primer i crucial pas en el desenvolupament d'una ontologia efectiva. Aquesta fase s'ocupa de respondre les preguntes: "¿Sobre què tractarà la nostra ontologia?" i "¿Quin abast tindrà?".

La nostra ontologia se centrarà en el domini específic de les novel·les de ficció. Per determinar la cobertura, hem considerat dos factors clau:

- **Rang d'Edat del Lector:** Hem identificat quatre categories d'edat per als lectors:

1. *Baby*: de 0 a 4 anys
2. *Children*: de 7 a 14 anys
3. *Teenager*: de 15 a 18 anys
4. *Adult*: més de 18 anys

Aquests rangs d'edat ens permetran adaptar les recomanacions segons les preferències i les necessitats de cada grup d'edat, assegurant que el contingut sigui adequat i rellevant.

- **Hàbit de Lectura:** També considerem si el lector té un hàbit de lectura establert. Aquesta informació és vital perquè ens permet adaptar les recomanacions a aquells que són lectors habituals i a aquells que estan buscant iniciar-se en el món de la lectura o són mes principiants.

Amb aquests criteris establerts, podem assegurar que l'ontologia aborda de manera efectiva les necessitats i preferències dels usuaris dins del domini establert. També ens ajuda a delimitar la informació que s'incorporarà a l'ontologia, optimitzant la seva utilitat i eficiència.

En resum, aquesta fase ens ha proporcionat una base sòlida i clara sobre la qual construïrem les fases posteriors del desenvolupament de l'ontologia. Definir clarament el domini i la cobertura ha estat essencial per garantir que l'ontologia sigui coherent, útil i alineada amb les necessitats del nostre sistema de recomanació.

Fase 2: Considerar la reutilització d'ontologies existents

En el procés de desenvolupament d'ontologies, la reutilització d'ontologies existents pot oferir avantatges significatius com ara l'estalvi de temps i l'aprofitament d'esforços previs. No obstant això, en el context del nostre SBC de recomanació de llibres, hem decidit no fer reutilització d'ontologies existents.

La principal raó d'aquesta decisió ha estat el fet que partíem des de zero, sense ontologies previes relacionades amb el nostre projecte específic. A més, volíem assegurar-nos que la nostra ontologia estigués totalment alineada amb les necessitats específiques del nostre sistema i que tingués una estructura i semàntica clarament adaptada al nostre domini.

Aquesta decisió, tot i que pot semblar que incrementa la càrrega de treball inicial, ens ha proporcionat una major llibertat i flexibilitat en el disseny de l'ontologia, permetent-nos modelar-la exactament d'acord amb les nostres necessitats i expectatives.

És important destacar que, tot i no haver reutilitzat ontologies existents, vam revisar algunes ontologies relacionades per inspirar-nos, com per exemple "auto.clp". Aquesta revisió preliminar ens va ser útil per a la conceptualització inicial i la identificació de possibles reptes.

Per tant, la construcció d'una ontologia des de zero, tot i que pot ser més laboriosa, ens ha permès personalitzar i ajustar l'ontologia a les especificitats del nostre projecte, garantint la seva precisió, coherència i adaptabilitat al context del nostre SBC de recomanació de llibres.

Fase 3: Enumerar els termes importants a l'ontologia

Un dels passos crucials en el desenvolupament d'una ontologia és identificar i enumerar els termes importants que es tindran en compte dins l'ontologia. Aquests termes representen les entitats, propietats i relacions que es consideraran i que formaran part de l'estructura bàsica de l'ontologia.

A partir de l'enunciat inicial del projecte, hem començat a extreure els termes que consideràvem fonamentals per a la nostra ontologia. Aquesta tasca inicial d'enumeració va ser posteriorment enriquida i ampliada mitjançant la consulta de diverses fonts d'informació. Aquestes fonts ens han permès identificar altres termes que no estaven explícitament mencionats a l'enunciat però que eren crucials per a l'ontologia.

A continuació es presenta una llista dels termes identificats:

1. Lector	7. name_genre	13. famous	19. goal
2. Autor	8. subgenre_of	14. name_author	20. hab_lectura
3. Libro	9. fav_author	15. public	21. illustrated
4. Ficción	10. author_of	16. has_subgenre	22. language
5. Genero	11. main_genre	17. bestseller	23. saga
6. Subgenero	12. actual	18. cover	24. theme

25. time_read	27. difficulty	29. punctuation	31. title
26. triggerw	28. length	30. rating	

Cada un d'aquests termes jugarà un paper fonamental en la configuració de la nostra ontologia i serà essencial per a la modelització de les relacions i propietats que determinaran com es genera la recomanació de llibres. És vital destacar que aquesta llista pot ser dinàmica i s'ajustarà segons les necessitats que vagin sorgint durant el desenvolupament del SBC.

Fase 4: Classes y Jerarquia

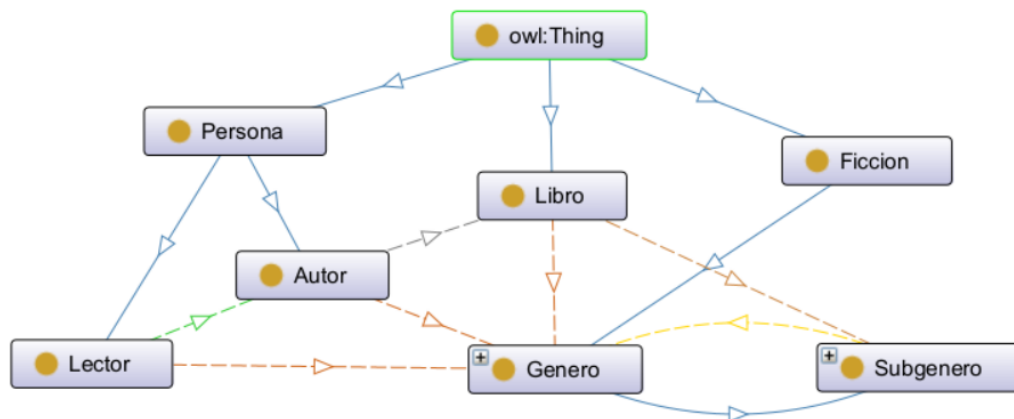


Figure 2: Jerarquia de Classes

En aquesta imatge podem observar la representació de totes les classes generades juntament amb les seves relacions. En primer lloc trobem la classe "Persona", la qual es divideix en dues subclasses: "Lector" i "Autor". Malgrat que no s'utilitza la classe "Persona" de forma directa en cap situació, les dues subclasses comparteixen atributs similars i comparteixen la característica fonamental de ser éssers humans. Per aquest motiu, s'ha optat per agrupar-les sota aquest concepte comú.

Al centre de la imatge, trobem la classe central d'aquest sistema, la classe "Llibres". Tot i que aquesta classe no conté cap subclasse específica, està interconnectada amb la majoria de les altres classes del sistema, ja que serveix com a punt central on es recopilen i gestionen les dades rellevants sobre els llibres.

Desplaçant-nos cap a la dreta, trobem la classe "Ficció", que a la vegada inclou la classe "Gènere" i, aquesta última, conté la classe "Subgènere". Aquestes classes no disposen d'atributs directes, sinó que es creen instàncies de cada classe per fer referència a alguns atributs relacionats amb autors, lectors i llibres. Així mateix, la classe "Ficció" es presenta sense atributs o instàncies pròpies, però té una importància conceptual clara ja que serveix com a classificació principal per als gèneres que s'apliquen als llibres de ficció.

Aquesta representació de classes i les seves relacions proporciona una estructura ben definida i entenedora pel sistema, facilitant la gestió de dades.

Fase 5: Propietats de las classes i Fase 6: Característiques de las propietats

En aquesta fase, es detallen les propietats de les classes, així com les seves característiques. Aquesta definició és fonamental ja que determina la manera en què les classes interactuen entre si, així com la informació específica que contenen.

Els atributs són definits amb el seu tipus corresponent i, quan es pertanyent, els valors que poden prendre. A continuació, es detallen les propietats de les classes més rellevants:

- **Autor:**

- actual: SYMBOL - Indica si l'autor és actual.
- famous: SYMBOL - Representa si l'autor és reconegut a nivell mundial.
- name.author: STRING - Nom complet de l'autor.
- public: STRING - Tipus de públic al qual es dirigeix principalment l'autor (edat).

- **Lector:**

- actual: SYMBOL - Indica si el lector prefereix llibres d'actualitat.
- bestseller: SYMBOL - Prefereix llibres que siguin bestsellers.
- cover: STRING - Preferències relacionades amb el tipus de portada del llibre.
- famous: SYMBOL - Interès en llibres d'autors famosos.
- goal: STRING - Objectiu principal de la lectura (entreteniment, aprenentatge, etc.).
- hab.lectura: SYMBOL - Hàbits de lectura del lector.
- illustrated: SYMBOL - Preferència per llibres il·lustrats.
- language: STRING - Idioma en què prefereix llegir el lector.
- public: STRING - Tipus de públic amb el qual el lector es identifica.
- saga: STRING - Interès en sagues o col·leccions de llibres.
- theme: STRING - Temàtica de preferència del lector.
- time.read: STRING - Temps que dedica el lector a la lectura.
- triggerw: STRING (multislot) - Temes que el lector prefereix evitar.

- **Libro:**

- actual: SYMBOL - Indica si el llibre és d'actualitat.
- bestseller: SYMBOL - Indica si el llibre ha estat un bestseller.
- cover: STRING - Tipus de portada del llibre.
- difficulty: STRING - Nivell de dificultat de lectura del llibre.
- goal: STRING - Objectiu principal del llibre (informar, entretenir, etc.).
- illustrated: SYMBOL - Indica si el llibre està il·lustrat.
- language: STRING - Idioma original del llibre.
- length: STRING - Longitud o número de pàgines del llibre.
- public: STRING - Públic objectiu del llibre.
- punctuation: INTEGER - Puntuació mitjana del llibre.
- rating: FLOAT - Valoració general del llibre.
- saga: STRING - Pertinença a una saga o col·lecció.
- theme: STRING - Temàtica principal del llibre.
- title: STRING - Títol del llibre.
- triggerw: STRING (multislot) - Temes que poden ser sensibles dins del llibre.

Cal destacar que, per exemple, l'atribut "triggerw" s'identifica com a "multislot", indicant que pot contenir múltiples valors dins de la seva definició.

Fase 7: Creació d'instàncies

Durant el procés de formalització, la creació d'instàncies esdevé essencial per donar sentit i aplicació pràctica a la nostra ontologia. Hem seguit diferents procediments per crear instàncies:

1. **Instàncies de gènere i subgènere creats a Protégé:** Les instàncies de gènere i subgènere van ser definides directament dins de l'entorn de Protégé:

- Romàntic (Romance)
 - Romàntic Clàssic (Classic_Romance)
 - Eròtic (Erotic)
 - Romàntic Històric (Historical_Romance)
- Contemporani (Contemporary)
 - Romàntic Històric Contemporani (Historical_Romance_Contemporary)
 - Distopia Real (Real_Dystopia)
 - Ficció Realista (Realistic_Fiction)
- Ciència Ficció (Science_Fiction)
 - Cyberpunk
 - Distopia Postapocalíptica (Postapocalyptic_Dystopia)
 - Òpera Espacial (Space_Opera)
 - Tecnològic (Technological)
- Fantasia (Fantasy)
 - Fantasia Obscura (Dark_Fantasy)
 - Alta Fantasia (High_Fantasy)
 - Fantasia Urbana (Urban_Fantasy)
- Històric (Historical)
 - Història Fictícia (Fiction_History)
 - Realista (Realistic)
- Horror (Horror)
 - Horror Paranormal (Paranormal_Horror)
 - Horror Psicològic (Psychological_Horror)
 - Thriller Horror (Thriller_Horror)
- Misteri (Mystery)
 - Misteri Paranormal (Paranormal_Mystery)
 - Policial (Police)
 - Thriller Misteri (Thriller_Mystery)
- Distopia (Dystopia)
- Social (Social)
- Aventura (Adventure)

2. **Instàncies d'autor i llibre creats amb base de dades:** Per a les instàncies relacionades amb autors i llibres, les implementarem en un futur amb la creació de una base de dades pròpia.

Ontologia final

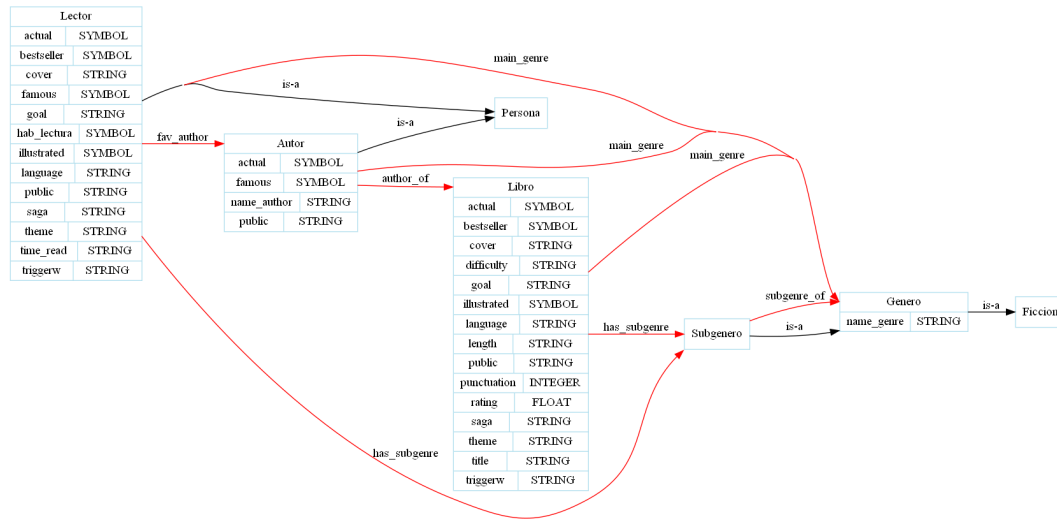


Figure 3: Classes, atributs i relacions

5.3 Mètode de Resolució

Com ja s'ha detallat a la secció 4 de Conceptualització, aquest problema es centra principalment en tasques d'anàlisi. Malgrat això, es pot dividir en dos eixos fonamentals, un d'**anàlisi** i l'altre de **síntesi**. En aquesta secció es descriuran els mètodes que utilitzarem per resoldre cadascun d'aquests eixos del problema.

L'eix que es centra en comprendre les preferències del lector és de naturalesa **analítica** i es resoldrà fent preguntes amb respostes tancades. Cada resposta obtinguda s'anirà guardant en els diferents atributs de la classe "lector" per a posterior utilització. A més, no totes les preguntes seran iguals per a tothom. Es faran preguntes tenint en compte les respostes de preguntes prèvies.

D'altra banda, l'eix que es dedica a la recomanació de llibres amb la informació obtinguda sobre les preferències del lector és una tasca de síntesi. Un cop hem definit les classes i els seus atributs, hem identificat que hi ha característiques més restrictives que altres. És a dir, no totes les característiques tenen el mateix pes en la recomanació.

Una classificació heurística s'adaptaria al nostre format ja que podríem donar més puntuació a aquells atributs que són més rellevants i menys puntuació als menys importants. Tot i això, hem decidit complementar aquest mètode de resolució degut a que existeix una petita probabilitat que s'acabi triant un llibre incorrecte si coincideix exactament amb totes les característiques del lector menys la característica més important. La suma de les puntuacions de totes les característiques coincidents pot arribar a ser més gran que la puntuació de la característica important.

Per aquest motiu creiem que és millor optar per fer una combinació de classificació heurística i mètode de ramificació i poda.

Mètode de Ramificació i Poda

Després d'adquirir coneixement expert, hem arribat a la conclusió que els atributs 'public' i 'Trigger_Warnings' tenen una importància destacada en comparació amb els altres. És fonamental assegurar que el llibre s'adeqüi a l'edat del lector i que no contingui temes que aquest desitja evitar. Per tant, en aquests casos, al rebre les respostes de les preguntes, es realitza una eliminació automàtica de tots els llibres que no coincideixen amb el grup d'edat del lector o que contenen contingut explícit que el lector vol evitar.

Aquesta estratègia no només assegura una millor adaptació dels llibres a les preferències del lector, sinó que també redueix el conjunt de llibres disponibles per a la recomanació. Com a resultat, es simplifica

la tasca de classificació heurística, ja que hi ha menys llibres per a sumar-li puntuació i, per tant, es redueix la complexitat computacional associada a aquest procés.

Classificació Heurística

Amb la resta dels atributs utilitzarem el mètode de classificació heurística mitjançant un mètode de puntuació. Un cop haguem rebut totes les respostes a les preguntes i, per tant, disposem de tota la informació necessària sobre el lector, procedirem a avaluar els llibres basant-nos en aquesta informació. Examinarem els atributs de cada llibre i els compararem amb els atributs del lector. Si hi ha coincidències, se sumarà un valor a la puntuació del llibre.

A més, hem establert un criteri addicional per resoldre el problema de l'empat en situacions en les quals diversos llibres obtinguin la mateixa puntuació després de l'avaluació. A tots els llibres se'ls sumarà el valor de l'atribut 'rating', que representa la valoració o puntuació d'un llibre. D'aquesta manera, si hi ha llibres amb puntuacions iguals, serà el llibre amb la millor valoració el que sortirà com a recomanació.

Finalment, el sistema recomanarà els tres llibres amb les puntuacions més altes al usuari, assegurant una selecció de llibres que millor s'ajusti a les seves preferències.

6 Creació de la Base de Dades Sintètica

Remuntant-nos a l'apartat 3, vam arribar a la conclusió que, per no temptar a cometre errors a nivell d'ús de dades protegides i alhora poder ensenyar el potencial del nostre sistema recomanador, era millor optar per l'opció de crear les dades sintèticament.

Aclariment Ètic

Abans de tot ens agradaria resaltar la **no-existència** de les dades que utilitzarem en aquest treball. Tots els noms d'autor i llibres estan **generats artificialment**, al igual que les propietats.

L'**estructura** de les dades ha estat pensada per tots els integrants del grup i no ha pretès copiar cap mètode d'estructuració de cap companyia o recurs web.

Per tal que la base de dades s'aproximi el més possible a semblar-se a una de real a nivell de propietats i distribucions ens hem recolzat en la justificació en base al coneixement expert adquirit mitjançant l'entrevista i la cerca d'informació per internet.

Hem intentat en tot moment trobar les fonts amb més rigurositat, tot i que hi ha hagut algunes qüestions en què no ho hem pogut validar. En aquests casos hem optat per, si té mitjana lògica, generar la distribució uniforme aleatòriament.

En definitiva, hem intentat representar al màxim possible les distribucions reals d'un catàleg d'una llibreria i que, alhora, ens permetin jugar i investigar tot l'espai de possibilitats que engloben les característiques d'un llibre i els seus autors.

6.1 Generador de Dades

No volem entrar massa en detall en l'script usat per a la generació de la base de dades però el comentarem per a què es pugui entendre la idea i les distribucions usades.

Primer de tot constàvem d'un nombre molt gran de noms generats per la API d'OpenAI que dona ple consentiment d'ús i propietat de la generació.

Aquests noms estan organitzats en categories per edat i per gènere (en el cas dels adults i joves) o temàtica (en el cas dels nens i els petits). Vam haver de crear una funció (`isalphareal()`) que comprovava que els caràcters fossin lletres romàniques i una altra (`remove_accents_and_n()`) per a treure accents i "ñ".

També vam definir abans de la creació les classes **Book** amb els seus atributs i **Author** amb els seus respectivament.

Veiem ara la producció i creació d'autors i llibres.

```
def generateData(
    public_probs:dict,
    adult_genres_probs:dict, # Clau: gènere, Valor: probabilitat
    adult_subgenres:dict, # Arbre de subgèneres d'adults invertit
    adol_genres_probs:dict,
    adult_names:dict,
    teenager_names:dict,
    children_names:dict,
    baby_names:dict,
    triggerw:dict,
    triggerw_probs_per_subgenre:dict,
    genre_teen_trigg_prob:dict,
    num_authors = 100
) -> Tuple[pd.DataFrame]:
```

Aquí podem veure la definició de la funció que pren com a paràmetres les probabilitats de cada propietat (públic, gènere, etc.), la jerarquia de gèneres, els noms pels llibres de cada públic i els *trigger warnings* amb les seves respectives probabilitats en base al gènere i subgènere.

En relació a les distribucions de probabilitat, hem tingut en compte la informació procedent de dues fonts: la Web1 i la Web2. La primera font ens proporciona dades sobre els llibres venuts, mentre que la segona font ofereix informació sobre els llibres llegits pels usuaris. Mitjançant la combinació d'aquestes dues fonts de dades, som capaços d'obtenir una idea més acurada dels percentatges de llibres a tenir. A continuació mostrem un gràfic per cadascuna de les distribucions.

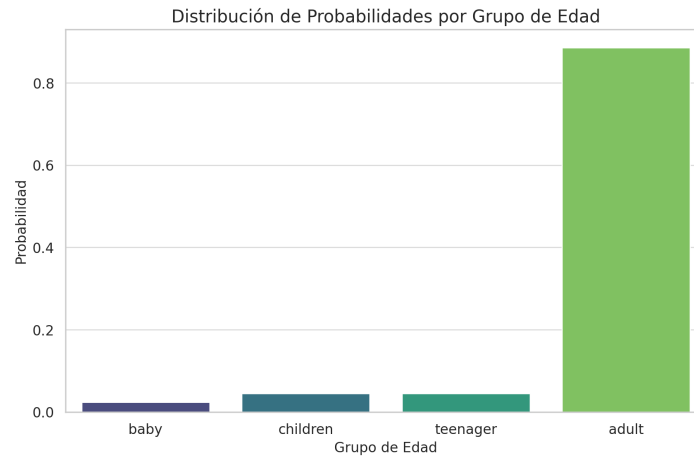


Figure 4: Probability Distribution of *public*

En les dues fonts d'informació s'assenyala que el gran gruix de llibres venuts i llegits són del públic adult, i per tant aquesta distribució és proporcional a la realitat.

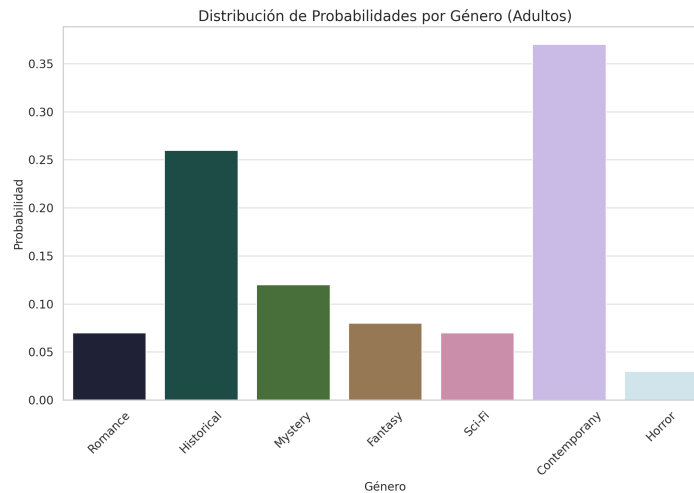
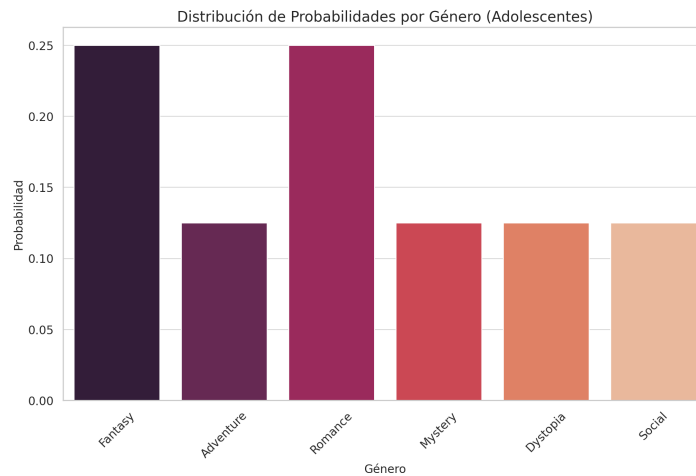


Figure 5: Probability Distribution of *genres* (adult)

En aquest cas, la Web2 presenta informació més acurada sobre els percentatges dels llibres venuts segons els gèneres. Aquestes distribucions per tant, coincideixen amb la informació obtinguda per la web esmentada, on a la majoria d'adults prefereixen un llibre de gènere contemporani, que engloba totes aquelles temàtiques més actuals.

Figure 6: Probability Distribution of *genres* (teen)

De la mateixa manera que en el gràfic anterior, també s'han creat les distribucions basant-nos en la informació extreta per la pàgina Web2. En aquest cas, els adolescents prefereixen principalment llibres romàntics o de fantasia.

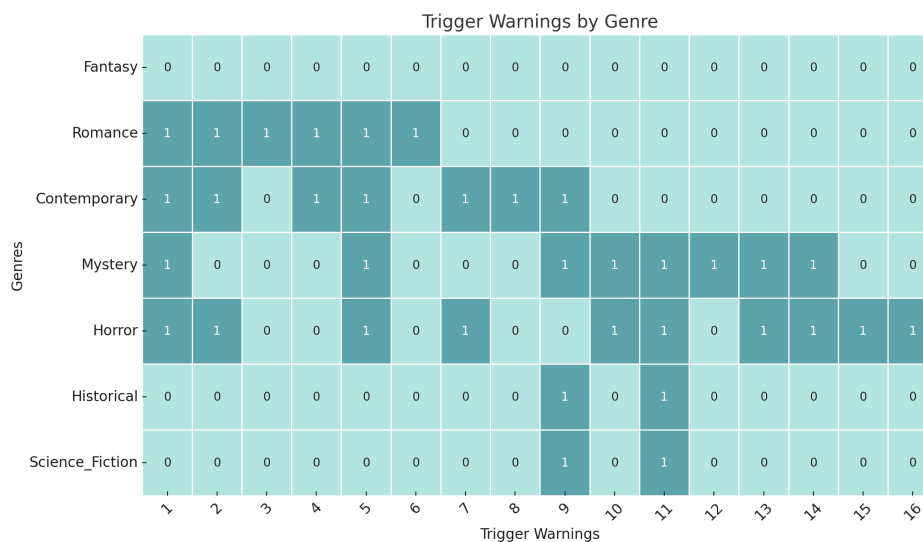


Figure 7: Trigger Warnings by Genre

Per últim, la decisió d'incorporar els "trigger warnings" va sorgir de la entrevista amb l'experta en el camp, que va especificar quins gèneres podien contenir aquests temes delicats i quins no. A més, va proporcionar una llista dels "trigger warnings" més comuns. Amb el coneixement expert que hem adquirit, vam prendre la decisió de determinar quins "trigger warnings" estan associats a cada gènere. Aquesta col·laboració amb l'experta i la nostra pròpia perspectiva experta van ser crucials per a establir aquesta característica en el sistema.

Un cop visualitzades les distribucions de probabilitat, veiem com hem dut a terme la creació d'autors:

```
# Creem els autors
for i in range(num_authors):
    print(f'Author {i}')
    public = np.random.choice(list(public_probs.keys()),
                              p=list(public_probs.values()))
    isactual = np.random.choice([True, False])
```

```

name = createname()
if public == 'adult':
    genre = np.random.choice(list(adult_genres_probs.keys()),
                             p=list(adult_genres_probs.values()))
elif public == 'teenager':
    genre = np.random.choice(list(adol_genres_probs.keys()),
                             p=list(adol_genres_probs.values()))
else:
    genre = None
author = Author(name,isactual,public,genre)
authors_object.append(author)

```

El nom, com es pot apreciar, és fruit de la funció `createname()`, que es connecta a una API pública generadora de noms i genera un nom que compleixi les característiques de no contenir cap caracter estrany. La funció és la següent:

```

def createname() -> str:
    url = "https://randomuser.me/api/"
    try:
        r = True
        while r:
            response = requests.get(url)
            response.raise_for_status() # Lanza un error si la solicitud no fue exitosa
            data = response.json()
            user = data['results'][0]
            if isalphareal(user['name']['first']) and isalphareal(user['name']['last']):
                name = user['name']['first'] + ' ' + user['name']['last']
                r = False
        return name
    except requests.exceptions.RequestException as e:
        print(f"Error en la solicitud: {e}")

```

Després es creen els llibres. Com que les seves propietats depenen del públic, hem dividit la creació en quatre blocs. Explicarem el primer d'ells, el d'adults, que és el més complet i amb més propietats amb probabilitats:

```

books = []

# Llibres d'adults
for sg in list(adult_names.keys()):
    for title in adult_names[sg]:
        title = remove_accents_and_n(title)
        name = title
        subgenre = sg
        genre = adult_subgenres[sg]
        isactual = np.random.choice(['FALSE', 'TRUE'])
        public = 'adult'
        author_options = findauthors(authors_object, public, genre, isactual)
        if author_options == []:
            author = Author(createname(),isactual,public,genre)
            authors_object.append(author)
        else:
            author = np.random.choice(author_options)
        cover = np.random.choice(['pocket', 'soft', 'hard'])
        length = np.random.choice(['short', 'medium', 'long', 'very_long'],
                                   p=[0.15,0.35,0.35,0.15])
        bestseller = np.random.choice(['TRUE', 'FALSE'],p=[0.05,0.95])
        if bestseller:

```



```

    if isinstance(author, Author):
        author.famous = 'TRUE'
    else:
        author['famous'] = 'TRUE'
    rating = random.uniform(1,10)
    prob_tw = triggerw_probs_per_subgenre[genre][subgenre]
    tw = np.random.choice(['TRUE', 'FALSE'], p = [prob_tw, 1-prob_tw])
    if tw == 'TRUE':
        triggers = random.sample(triggerw[genre], random.randint(1,
            len(triggerw[genre])-1))
        trigger_warning = "{"
        for t in triggers:
            trigger_warning += ' '
            trigger_warning += t
        trigger_warning += '}'
    else:
        trigger_warning = ""
    book = Book(name, isactual, length, public,
        cover, genre, author, rating, subgenre=subgenre,
        bestseller=bestseller, triggerw=trigger_warning)
    books.append(book.__dict__)
    author.books.append(book)

```

La idea és anar iterant per cada nom de llibre de cada subgènere. Un cop tenim un títol assignat, identifiquem el subgènere, per conseqüència també el gènere, el públic i decidim aleatòriament si el llibre és actual o no amb equiprobabilitat.

Amb aquestes propietats, ja podem buscar un autor que compleixi amb aquestes característiques tant de públic com de gènere com d'actualitat. Per això usem la funció `findauthors()` que filtra i retorna un conjunt d'autors possibles.

```

def findauthors(
    authors,
    public,
    genre,
    isactual
) -> List['Author']:
    subset = []
    for a in authors:
        if a.public == public and a.genre == genre and a.isactual == isactual:
            subset.append(a)
    return subset

```

Després d'assignar un autor congruent, es genera un tipus de tapa, la longitud del llibre i si és *best-seller*. D'acord amb aquesta última propietat, farem que l'autor que ha escrit el llibre es converteixi automàticament en famós si no ho era ja abans.

Un cop fet això, assignem un valor al *rating* entre l'interval [1,10] uniformement distribuït. Aquesta propietat ens serà d'ajuda a l'hora d'ordenar llibres amb les mateixes propietats (i per tant puntuació).

Finalment, amb les probabilitats del gènere assignat decidim si el llibre conté temes que poden fer saltar algun trigger warning i, en base al subgènere, decidim quins.

Hi ha algunes propietats com *goal* o *theme* que no s'assignen ja que un llibre per adults no té aquestes característiques en la nostra representació del domini. Aquestes, en canvi, sí que s'assignaran en altres tipus de public.

6.2 Conversió de les Instàncies

Per a la conversió de totes les instàncies d'autor i llibre generats amb l'arxiu `generator.py` hem creat un altre arxiu (`generador_instancias.py`) que pren les dues bases de dades en `.csv`. Aquest itera per les dues i genera un (`definstances instances`) amb cadascun dels llibres i autors creats que després podem importar durant l'execució del recomanador.

Per aquest motiu, caldria executar el següent abans de començar l'execució (**run**):

```
CLIPS> (load main.clp)
+#####@!!!!!!!!++++*****!!****++**+++***+*****+*****!****+***
TRUE
CLIPS> (load instances.clp)
@@
TRUE
CLIPS> (reset)
```

7 Implementació

7.1 Representació de l'Ontologia

En l'apartat anterior hem pogut veure la representació formal de l'ontologia amb la seva jerarquia de classes, les respectives intàncies explícites, els atributs de cada classe i les relacions entre elles.

Ara doncs veurem de manera senzilla com s'ha dut a terme l'exportació de l'ontologia `.ttl` a un arxiu de CLIPS amb totes les classes definides de manera correcta.

Per a aconseguir una exportació de manera senzilla per a nosaltres se'ns va brindar un arxiu `owl2clips` a partir de la utilització de la llibreria `owl2else` que et duia a terme la conversió. Aquest amb la simple comanda

```
$ python owl2clips.py --input ontology.ttl --format turtle
```

és capaç de convertir l'ontologia a format `.clp`.

Com veurem a continuació, les definicions de totes les classes i instàncies explícites les hem posades a l'inici del mòdul MAIN de tots els nostres prototips fins al final ja que conformen l'esquelet del nostre sistema expert.

Anem poc a poc comentant la conversió:

```
(defclass Ficcion
  (is-a USER)
  (role concrete)
  (pattern-match reactive)
)

(defclass Genero
  (is-a Ficcion)
  (role concrete)
  (pattern-match reactive)
  (slot name_genre
    (type STRING)
    (create-accessor read-write))
)

(defclass Subgenero
  (is-a Genero)
  (role concrete)
  (pattern-match reactive)
  (slot subgenre_of
    (type INSTANCE)
    (create-accessor read-write))
)
```

En aquest primer tast ja podem observar la creació de tota la jerarquia per la banda dels gèneres. Amb Ficción com a classe pare, Genero com a classe que hereda de Ficción i Subgenero com a subclasse de Genero. A més a més ens podem fixar en la propietat `name_genre` de Genero (i per tant de la seva classe filla) que es defineix com a `STRING` tal i com s'ha decidit a l'ontologia.

També cal destacar l'aparició de la primera relació entre classes (`subgenre_of`) que apareix en format d'*slot* de tipus `INSTANCE`.

```
(defclass Persona
  (is-a USER)
  (role concrete)
  (pattern-match reactive)
)
```

```

(defclass Autor
  (is-a Persona)
  (role concrete)
  (pattern-match reactive)
  (multislot author_of
    (type INSTANCE)
    (create-accessor read-write))
  (slot main_genre
    (type INSTANCE)
    (create-accessor read-write))
  ...
  (slot public
    (type STRING)
    (create-accessor read-write))
)

(defclass Lector
  (is-a Persona)
  (role concrete)
  (pattern-match reactive)
  (slot fav_author
    (type INSTANCE)
    (create-accessor read-write))
  (slot has_subgenre
    (type INSTANCE)
    (create-accessor read-write))
  (slot main_genre
    (type INSTANCE)
    (create-accessor read-write))
  ...
  (multislot triggerw
    (type STRING)
    (create-accessor read-write))
)

```

En aquest segon fragment veiem tota la branca de la jerarquia de classes de **Persona**, una classe merament auxiliar per a l'agrupament de dues entitats crucials pel desenvolupament del sistema: **Lector** i **Autor**.

L'autor, com s'ha mencionat a l'apartat de formalització conté els següents atributs i relacions amb altres classes:

- **author_of** (multislot, INSTANCE)
- **main_genre** (slot, INSTANCE)
- **actual** (slot, SYMBOL)
- **famous** (slot, SYMBOL)
- **name_author** (slot, STRING)
- **public** (slot, STRING)

De mateixa forma podem fer pel lector:

- **fav_author** (slot, INSTANCE)
- **has_subgenre** (slot, INSTANCE)
- **main_genre** (slot, INSTANCE)
- **actual** (slot, SYMBOL)

- bestseller (slot, SYMBOL)
- cover (slot, STRING)
- famous (slot, SYMBOL)
- goal (slot, STRING)
- hab_lectura (slot, SYMBOL)
- illustrated (slot, SYMBOL)
- language (slot, STRING)
- public (slot, STRING)
- saga (slot, STRING)
- theme (slot, STRING)
- time_read (slot, STRING)
- triggerw (multislot, STRING)

Finalment, anem a veure la definició de la classe Libro.

```
(defclass Libro
  (is-a USER)
  (role concrete)
  (pattern-match reactive)
  (slot has_subgenre
    (type INSTANCE)
    (create-accessor read-write))
  (slot main_genre
    (type INSTANCE)
    (create-accessor read-write))
  (slot actual
    (type SYMBOL)
    (create-accessor read-write))
  ...
  (slot title
    (type STRING)
    (create-accessor read-write))
  (multislot triggerw
    (type STRING)
    (create-accessor read-write))
)
```

Aquesta, com ja s'ha comentat anteriorment, és la classe amb més atributs diferents. Aquí deixem un llistat amb els seus respectius tipus:

- has_subgenre (slot, INSTANCE)
- main_genre (slot, INSTANCE)
- actual (slot, SYMBOL)
- bestseller (slot, SYMBOL)
- cover (slot, STRING)
- difficulty (slot, STRING)
- goal (slot, STRING)
- illustrated (slot, SYMBOL)
- language (slot, STRING)

- length (slot, STRING)
- public (slot, STRING)
- punctuation (slot, INTEGER)
- rating (slot, FLOAT)
- saga (slot, STRING)
- theme (slot, STRING)
- title (slot, STRING)
- triggerw (multislot, STRING)

Noti's que els *triggerw* estan representats en un multislot ja que no hi ha un únic valor possible per cada llibre. A més a més els llibres consten de dues relacions entre classes, una amb els gèneres (**main_genre**) i l'altra amb el subgènere específic (**has_subgenre**).

L'últim a mencionar son les **instàncies específiques** de gènere i subgènere. Aquestes les vam definir manualment, amb el seu nom des de Protégé i les vam exportar juntament amb les classes.

A més a més també vam crear la instància de **Lector**, que serà l'actor principal en la resolució del problema. Aquest comença buit, amb totes les qualitats a emplenar per l'usuari.

```
(definstances instances
  ([Adventure] of Genero
    (name_genre "Adventure")
  )

  ([Classic_Romance] of Subgenero
    (subgenre_of [Romance])
    (name_genre "Classic_Romance")
  )

  ([Contemporary] of Genero
    (name_genre "Contemporary")
  )

  ([Cyberpunk] of Subgenero
    (subgenre_of [Science_Fiction])
    (name_genre "Cyberpunk")
  )
  ...
  ([Global] of Genero
    (name_genre "Global")
  )
  ...
  ([Urban_Fantasy] of Subgenero
    (subgenre_of [Fantasy])
    (name_genre "Urban_Fantasy")
  )

  ([lector] of Lector
    (triggerw "")
  )
)
```

Hem volgut destacar la instància auxiliar de **Genero** [Global] ja que no pertany a la modelització del problema sinó que sols ens és útil a nivell d'implementació.

Com explicarem ens el següents apartats sobre els mòduls i prototips, aquesta ens servirà per no deixar el valor de gènere buit després d'haver-li preguntat a l'usuari pel gènere de preferència i que aquest respongui que no en té (respongui '-').

7.2 Mòduls de Resolució

En aquest apartat, descriurem els diferents mòduls que hem implementat en el nostre SBC de recomanació de llibres i explicarem les raons darrere de la seva creació. La modularització del nostre sistema té diversos avantatges. En primer lloc, permet una organització més clara del codi. A més, podem separar amb facilitat les normes específiques depenent del grup d'edat al qual pertany el client. Això facilita el manteniment i l'extensió del sistema, ja que podem modificar o afegir regles en un mòdul específic sense afectar la resta. També ens ofereix la capacitat de cridar a la mateixa norma des de diferents perfils, proporcionant una major flexibilitat.

A continuació, es mostra un esquema que representa les relacions entre els mòduls del sistema. Les fletxes grises indiquen de quins mòduls provenen les crides cap a altres mòduls, amb números que indiquen l'ordre en què es realitzen les crides. Les fletxes vermelles mostren cap a quin mòdul es retorna després de completar l'execució del mòdul corresponent.

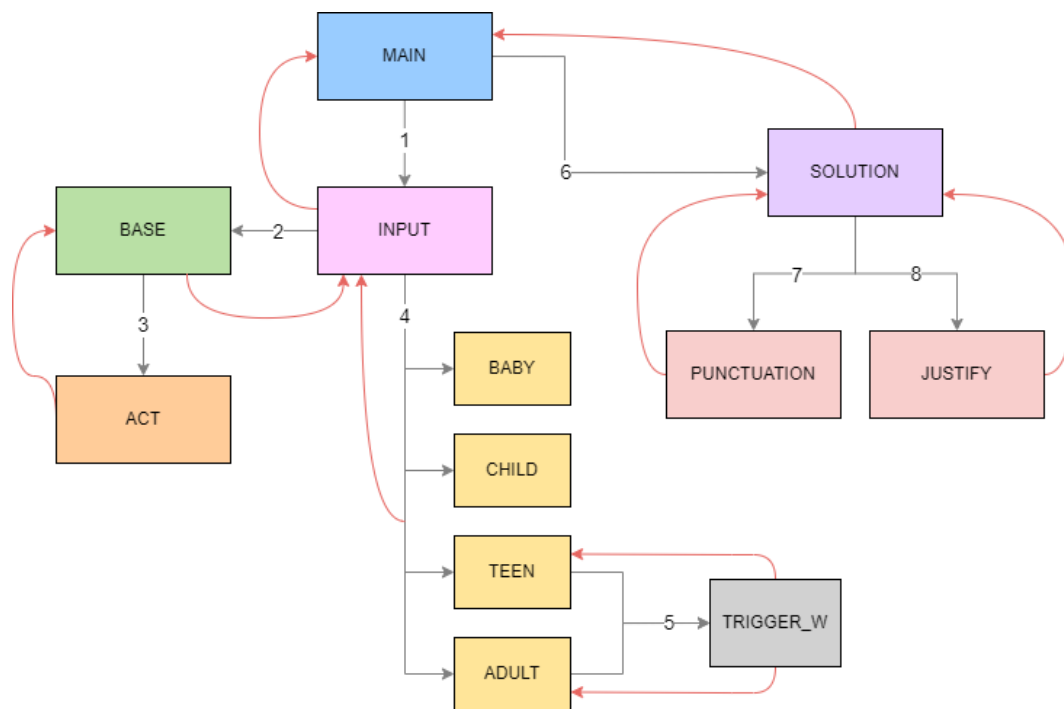


Figure 8: Relacions entre mòduls

7.2.1 Mòdul MAIN

Aquest mòdul actua com a punt d'entrada al sistema. La seva funció principal és obtenir l'input de l'usuari, processar aquesta informació i, finalment, retornar la solució al client.

Per a l'obtenció de tota la informació de cada mòdul hem executat les seüents comandes fent **focus** del mòdul que volem consultar:

```

1 CLIPS> (focus MAIN)
2 TRUE
3 CLIPS> (list-defrules)
4 benvinguda
5 init
6 end
7 For a total of 3 defrules.
8 CLIPS> (list-deffunctions)

```

```

9 ask-question
10 yes-or-no-p
11 multi-select-p
12 countInstances
13 For a total of 4 deffunctions.
14 CLIPS> (ppdefmodule MAIN)
15 (defmodule MAIN
16   (export ?ALL))

```

- **Imports:** -
- **Exports:** ?ALL
- **Reglas:** benvinguda, init, end
- **Funcions:** ask-question, yes-or-no-p, multi-select-p, countInstances

7.2.2 Mòdul INPUT

Dins d'aquest mòdul, es recopila informació relativa al perfil del lector.

- **Imports:** MAIN ?ALL
- **Exports:** deffunction basicQ, basicQskip
- **Reglas:** basicInput, cridaPublic
- **Funcions:** basicQ, basicQskip

7.2.3 Mòdul BASE

Aquest mòdul conté regles i procediments relacionats amb preguntes bàsiques que són comunes independentment del perfil del lector.

- **Imports:** MAIN ?ALL, INPUT ?ALL
- **Exports:** -
- **Reglas:** askPublic, askHabit
- **Funcions:** -

7.2.4 Mòdul BABY

Dins del mòdul BABY, es defineixen regles específiques per al perfil dels més petits. Aquí es formulen preguntes específiques per obtenir més informació sobre les preferències i necessitats d'aquest grup d'edat.

- **Imports:** MAIN ?ALL, INPUT deffunction basicQ, basicQskip
- **Exports:** -
- **Reglas:** atributs, tematica, goal
- **Funcions:** -

7.2.5 Mòdul CHILDREN

Similar al mòdul BABY, el mòdul CHILDREN conté regles específiques per als nens. Les preguntes formulades aquí estan dissenyades per recopilar informació detallada sobre el perfil i preferències d'aquest grup d'edat.

- **Imports:** MAIN ?ALL, INPUT deffunction basicQ, basicQskip
- **Exports:** -
- **Reglas:** atributs, tema_preferit, llibre_illustrat, longitud
- **Funcions:** -

7.2.6 Mòdul TEEN

En aquest mòdul, es desenvolupen regles específiques per als adolescents. Les preguntes es centren en temes i categories de llibres que podrien ser d'interès per a aquest grup d'edat.

- **Imports:** MAIN ?ALL, INPUT deffunction basicQ, basicQskip
- **Exports:** -
- **Reglas:** atributs, genere_teen, tipus_de_coberta, llibre_trendy, longitud
- **Funcions:** -

7.2.7 Mòdul ADULT

El mòdul ADULT conté regles específiques per als adults. Aquí, les preguntes es formulen tenint en compte les diferents preferències i temes que podrien interessar als lectors d'aquesta categoria.

- **Imports:** MAIN ?ALL, INPUT deffunction basicQ, basicQskip
- **Exports:** -
- **Reglas:** atributs, temps_de_lectura, askFamous, interes_bestseller, tipus_de_coberta, preferencia_novela, genere_adult, activaTriggerW, subgenere
- **Funcions:** subg-disp

7.2.8 Mòdul TRIGGER_W

Aquest mòdul està dedicat a obtenir informació sobre si el lector vol evitar certes temàtiques o continguts.

- **Imports:** MAIN ?ALL, INPUT deffunction basicQ, basicQskip
- **Exports:** -
- **Reglas:** trigger_avoid, podaTriggers
- **Funcions:** -

7.2.9 Mòdul ACT

Dins del mòdul ACT, es realitzen diferents accions sobre els atributs recopilats en els mòduls anteriors. Això podria incloure l'ordenació de les preferències, l'assignació de pesos a diferents categories, entre altres.

- **Imports:** MAIN ?ALL
- **Exports:** -
- **Reglas:** inici, updatePublic, podaPublic
- **Funcions:** updateP

7.2.10 Mòdul SOLUTION

El mòdul SOLUTION s'encarrega d'anunciar que s'ha arribat al final del procés de preguntes i cridar els mòduls que s'encarreguen de donar la solució final.

- **Imports:** MAIN ?ALL
- **Exports:** -
- **Reglas:** endMessage, justifica
- **Funcions:** -

7.2.11 Mòdul PUNCTUATION

Com ja diu el nom d'aquest mòdul, únicament s'encarrega de fer els càlculs de la puntuació de cada llibre segons les característiques del lector.

- **Imports:** MAIN ?ALL
- **Exports:** -
- **Reglas:** theme_punctuation, goal_punctuation, ilustrated_punctuation, length_punctuation, cover_punctuation, genres_punctuation, bestseller_punctuation, famous_punctuation, actual_punctuation, subgenre_punctuation
- **Funcions:** -

7.2.12 Mòdul JUSTIFY

Finalment, el mòdul JUSTIFY s'encarrega de proporcionar la solució basant-se en tota la informació processada anteriorment. Retorna la recomanació de llibre o llibres més adequats per al perfil del lector i la justificació de la tria. A més, dona una frase informativa sobre el grau de coincidència entre les característiques del llibre triat i les demanades pel lector.

- **Imports:** MAIN ?ALL
- **Exports:** -
- **Reglas:** justify
- **Funcions:** getRecommendationRange, printRecom, showProperties, compareBooks, bestbooks, tematica, genre

Som conscients de que el nostre sistema està basat en regles i per tant la presència de funcions hauria de ser minoritària respecte la presència de regles. Tot i això, no hem trobat manera de poder mostrar una recomanació amb la seva justificació de manera ordenada fent ús principalment de regles. Ha sorgit la idea de fer, tal i com es planteja a la puntuació, una regla per a cada atribut, però no és viable si volem donar un llibre seguit de la seva justificació; haver-ho implementat amb regles suposaria un resultat pitjor per a l'usuari. Per aquest motiu hem vist adient fer aquesta excepció i definir moltes funcions.

7.3 Mètode de Resolució

En aquesta secció es presentarà en detall el mètode de resolució adoptat pel nostre SBC. La representació del procés de resolució mitjançant regles està dissenyada per seguir el raonament establert per la metodologia de resolució de problemes que hem escollit.

Estratègia de Resolució mitjançant Regles Les regles són la base de la nostra solució. Hem establert les següents característiques per a les nostres regles:

- Les regles s'activen únicament quan són necessàries. Això assegura eficiència en el procés de resolució.
- La condició d'activació d'una regla és mínima i indispensable. Això simplifica el procés de determinació de quina regla s'ha d'activar en un moment donat.
- Les regles són simples per assegurar que són fàcils de seguir i d'entendre.
- A més, hem decidit crear una sèrie de regles més complexes per a casos específics on es requereix una anàlisi més detallada.

7.4 Prototips i Mètode Incremental

Als apartats anteriors s'ha explicat tot el conjunt de mòduls i les seves respectives regles que conformen el nostre sistema expert recomanador de llibres final. El desenvolupament, però, ha estat un **procés iteratiu** on hem anat progressant i creant prototips funcionals cada cop amb més complexitat i/o qualitat de resultats.

En aquest apartat, doncs, veurem el primer prototip base que ens donarà una **primera solució** que no es basa en cap de les preferències del lector (ja que encara no seran assignades). El segon serà capaç de **podar** el conjunt d'instàncies de llibres en base al públic al què pertany el lector. El tercer, a més a més, constarà de **preguntes específiques** per al lector d'acord amb el seu grup d'edat que després, amb el quart prototip serà capaç de recomanar amb **millor precisió** els llibres escollits. Finalment mencionarem els últims **detalls i millores** que s'han aplicat per al desenvolupament dels prototips finals.

7.4.1 Prototip 1 - Solució Bàsica

En el prototip inicial ens vam assegurar de ser capaços d'ordenar un conjunt d'instàncies en base a un criteri. Com que no comptàvem encara amb cap preferència del lector vam optar per usar l'atribut *rating* dels llibres que, com és de tipus `FLOAT` uniformement distribuït entre 1 i 10, permet ordenar perfectament els llibres sense necessitat de cap altre informació.

Per dur-ho a terme ens vam veure amb l'obligació de crear les dues funcions del mòdul `MAIN`, `bestbooks` (?n) i `compareBooks`(?b1 ?b2), que ordenen els llibres segons el nostre criteri i mostren els 3 primers de la llista.

Aquí no ens va caler l'ús de cap regla. Sols vam definir l'inici i el final del procediment amb les següents regles del mòdul principal, que seran els pilars del nostre sistema expert:

```
(defrule MAIN::init
  (declare (salience 1))
  =>
  (focus INPUT)
)

(defrule MAIN::end
  =>
  (bestbooks 3)
  (printout t "Fin de la Ejecucion." crlf)
)
```

A més a més, vam definir el mòdul `INPUT` que seria el primer en ser cridat amb *focus* un cop s'executés la primera regla:

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

Així doncs, si carregàvem l'arxiu principal i l'arxiu amb les instàncies de llibres i autors, podem executar-ho i que ens recomanés els 3 millors llibres segons el seu *rating*. Ja teniem el primer prototip funcional.

Exemple d'execució

```
CLIPS> (load minimain.clp)
+#####@!!!!*+!
TRUE
CLIPS> (load instances.clp)
@@
TRUE
CLIPS> (reset)
CLIPS> (run)
Title: El susurro de tus palabras | Rating: 9.99835769232047
Title: Laberinto de Pantallas Silenciosas | Rating: 9.99814772264363
Title: A traves de tus ojos | Rating: 9.98876349540352
Fin de la Ejecucion.
```

7.4.2 Prototip 2 - Poda per Edat

La segona meta que ens vam proposar va ser la capacitat de podar de la base de dades totes aquelles instàncies que no complien ser del mateix públic (petits, nens, joves i adults) que el lector.

Per això necessitàvem 3 característiques:

- Ser capaços de rebre el públic i hàbit de lectura del lector.
- Poder fer una modificació del públic si el lector té hàbit.
- Poder eliminar els llibres que no es corresponguin al grup d'edat (públic).

Si ens fixem, la primera formaria part del mòdul `INPUT` ja que pretèn obtenir informació. En canvi, les altres dues sols modifiquen i alteren la base de dades i instàncies. Per això aquestes havien d'anar al mòdul `ACT`, d'actuadors.

En aquest moment va ser l'hora de definir les regles `basicInput`, `askPublic` i `askHabit`, que alhora requerien d'una funció que fos capaç d'acceptar una pregunta i uns valors a escollir, com seria `basicQ` (Q de question). Aquesta última vam decidir afegir-la al mòdul `INPUT`, degut a què es tractava d'una funció que ens podria ser útil en molts moments on haguem de recollir les preferències del lector. En canvi, vam decidir crear el mòdul `BASE` (que el cridaríem des de la regla `basicInput`) per a guardar les dues preguntes inicials de grup d'edat i hàbit de lectura.

A més a més, ja es va deixar l'estructura de com seguiria la relació entre mòduls d'entrada per a poder especificar en base al públic del lector. Ho vam fer amb la regla `cridaPublic`, que sols s'activava quan aquest ja no fos nul.

L'apartat doncs d'input va tenir la següent forma:

```
(defmodule INPUT
  (import MAIN ?ALL)
  (export deffunction basicQ)
)

(deffunction INPUT::basicQ (?question ?options)
  (bind ?response (ask-question ?question ?options))
  (return ?response)
)

(defrule INPUT::basicInput
  (declare (salience 1))
  =>
  (focus BASE)
)

(defrule INPUT::cridaPublic
  ?lector <- (object (is-a Lector) (public ?p&~""))
  =>
  (switch ?p
    (case "baby" then (focus BABY))
    (case "children" then (focus CHILD))
    (case "teenager" then (focus TEEN))
    (default (focus ADULT))
  )
)

(defmodule BASE
  (import MAIN ?ALL)
  (import INPUT ?ALL)
)
```

```

(defrule BASE::askPublic
  (declare (salience 1))
  ?lector <- (object (is-a Lector) (public ""))
  =>
  ...
)

(defrule BASE::askHabit
  ?lector <- (object (is-a Lector) (hab_lectura nil))
  =>
  ...
  (focus ACT)
)

```

Com es pot apreciar en aquesta última regla del mòdul base, abans d'abandonar el mòdul base i que, per tant, s'executi la regla que anirà als mòduls específics, es fa una crida al mòdul ACT.

Això és per a què s'actualitzi (en cas de que faci falta) el públic al qual pertany el lector i que, un cop fet, es dugui a terme la poda corresponent.

El mòdul ACT en aquell moment constava de la funció auxiliar `updateP (?public ?lector)`, de la regla amb preferència `updatePublic` i de la regla `podaPublic`.

A més a més, va ser el moment en què vam crear una altra funció auxiliar en el mòdul MAIN anomenada `countInstances (?clase)` que ens permet comptar el nombre d'instàncies que té una classe determinada.

```

(defun MAIN::countInstances (?clase)
  (bind ?num-instances (length$ (find-all-instances ((?i ?clase)) TRUE)))
  (printout t "Hi ha " ?num-instances " instàncies de " ?clase "." crlf)
  (return ?num-instances)
)

```

Exemple d'execució En aquesta execució es pot apreciar com ja som capaços de recollir dades del lector, actualitzar el públic (de *baby* a *children*), de podar i, per conseqüència, de recomanar llibres més infantils.

```

CLIPS> (load "minimain2.clp")
+#####@!!!!*****!*****+!***
TRUE
CLIPS> (load "instances.clp")
@@
TRUE
CLIPS> (reset)
CLIPS> (agenda)
1      init: *
0      end: *
For a total of 2 activations.
CLIPS> (run)

```

```

A quin grup d'edat pertanyes?
Si tens entre 4 i 6 anys tria l'opció 'baby'
Si tens entre 7 i 14 anys tria l'opció 'children'
Si tens entre 15 i 18 anys tria l'opció 'teenager'
Si tens més de 18 anys tria l'opció 'adult'

```

```

>>> baby
Public assignat correctament.

```

```

Public introduit: baby

Tens habit de lectura? Es a dir, llegeixes en els teus temps lliures,
quan vas de viatge... de manera coninuada?
  Respon 'Yes' o 'No'.

>>> Yes
S'ha rebut correctament la teva resposta:
  TRUE

S'ha actualitzat el public al qual pertanys:
  Pertanys al public 'children'.

Hi ha 927 instancies de Libro.
Hi ha 60 instancies de Libro.

Title: Siesta en la Habitacion Nubosa | Rating: 9.89168531121605
Title: Safari por el Zoo Fabuloso | Rating: 9.77635748040592
Title: Cena en el Comedor Girasol | Rating: 9.67698707509204
Fin de la Ejecucion.

```

7.4.3 Prototip 3 - Regles Específiques

Un cop vam poder discernir entre públic petit, infantil, jove i adult, vam seguir progressant en l'addició de regles específiques per a cada grup d'edat en els mòduls **BABY**, **CHILD**, **TEEN** i **ADULT**. Tal i com s'ha vist anteriorment, el mòdul dels petits recull informació sobre quina és la finalitat de la lectura i quina és la temàtica que prefereix. Així doncs, vam crear les regles **goal** i **tematica** que ens recollirien les dades i modificarien les propietats corresponents del lector.

```

(defmodule BABY
  (import MAIN ?ALL)
  (import INPUT deffunction basicQ)
)

(defrule BABY::tematica
  (declare (salience 2))
  ?lector <- (object (is-a Lector) (theme ""))
  =>
  ...
)

(defrule BABY::goal
  (declare (salience 1))
  ?lector <- (object (is-a Lector) (goal ""))
  =>
  ...
)

```

Pel public *child*, les preguntes que hem determinat anteriorment que se'ls havia de fer eren aquelles relacionades amb la temàtica (igual que els petits), però a més a més si volen que el llibre contingui il·lustracions. Per això vam implementar les regles **tema-preferit** i **llibre-illustrat** que ens recollirien les dades i modificarien les propietats corresponents del lector.

```

(defmodule CHILD
  (import MAIN ?ALL)
  (import INPUT deffunction basicQ)
)

```

```

(defrule CHILD::tema_preferit
  (declare (salience 1))
  ?lector <- (object (is-a Lector) (theme ""))
  =>
  ...
)
...

```

De la mateixa manera vam crear les regles `genere_teen`, `tipus_de_coberta` i `llibre_trendy` per a recollir les preferències del lector en el cas que fos jove. Aquestes preguntaven pel gènere que més li interessava, pel tipus de coberta que prefereix i per si està buscant llibres que estiguin de moda actualment o no.

```

(defmodule TEEN
  (import MAIN ?ALL)
  (import INPUT deffunction basicQ)
)

(defrule TEEN::genere_teen
  (declare (salience 1))
  ?lector <- (object (is-a Lector) (main_genre [nil]))
  =>
  (bind ?options Fantasy Adventure Romance Mystery Dystopia Social)
  ...
)
...

```

Per acabar, els adults són el públic més gran i amb major quantitat de llibres i opcions. Per això també tenen un conjunt més gran de preguntes que poden ser-nos útils alhora de recomanar-li un llibre o un altre. Les preguntes que vam implementar en aquest prototip són:

- De quant temps disposes per llegir? `ADULT::temps_de_lectura`
- Busques un llibre de algun autor considerat famós? `ADULT::askFamous`
- T'interessa llegir best sellers? `ADULT::interes_bestseller`
- Quin tipus de coberta prefereixes? `ADULT::tipus_de_coberta`
- T'agraden les novel·les actuals? O prefereixes les classiques? `ADULT::preferencia_novela`
- Quin genere de llibre estas buscant? `ADULT::genere_adult`

Després del desenvolupament dels mòduls d'input específics, vam deixar constància de la nostra intenció de tenir un mòdul `SOLUTION` que s'encarregués del subproblema de calcular les puntuacions finals de cada llibre. D'aquesta manera, la regla `MAIN::end` no cridaria a `bestbooks` de manera directa, sinó que ho faria mitjançant el mòdul solució.

```

(defrule MAIN::end
  =>
  (focus SOLUTION)
  (printout t "Fin de la Ejecucion." crlf)
)

(defmodule SOLUTION
  (import MAIN ?ALL)
)

(defrule SOLUTION::resultat
  =>
  (bestbooks 3)
)

```

Exemple d'execució Noti's que els llibres mostrats com a resultat no compleixen amb cap de les característiques específiques del públic *children* que ha introduït el lector. Això s'implementarà més endavant, al prototip 5 (sistema de puntuació).

```
CLIPS> (load "minimain3.clp")
+#####@!!!!**+**+!*****+*****+*****+*****!**
TRUE
CLIPS> (load "instances.clp")
@@
TRUE
CLIPS> (reset)
CLIPS> (agenda)
1      init: *
0      end: *
For a total of 2 activations.
CLIPS> (run)

A quin grup d'edat pertanyes?
  Si tens entre 4 i 6 anys tria l'opcio 'baby'
  Si tens entre 7 i 14 anys tria l'opcio 'children'
  Si tens entre 15 i 18 anys tria l'opcio 'teenager'
  Si tens mes de 18 anys tria l'opcio 'adult'

>>> baby
Public assignat correctament.
Public introduït: baby

Tens habit de lectura? Es a dir, llegeixes en els teus temps lliures,
quan vas de viatge... de manera coninuada?   Respon 'Yes' o 'No'.

>>> Yes
S'ha rebut correctament la teva resposta:
TRUE

S'ha actualitzat el public al qual pertanyes:
Pertanyes al public 'children'.

Hi ha 927 instancies de Libro.
Hi ha 60 instancies de Libro.

Quines son les teves aficions?
  Si t'agrada l'esport i les activitats fisiques, selecciona 'Sports'.
  Si ets un amant de la musica i els concerts, selecciona 'Music'.
  Si prefereixes les histories sobre la vida quotidiana, selecciona
    'Daily_life'.
  Si t'agrada explorar i viure aventures, selecciona 'Adventures'.
  Si t'estimes la cultura i l'art, selecciona 'Culture'.
  Si t'agrada la magia i el mon del fantastic, selecciona 'Magic'.
  Si t'agrada el mon dels esports i la competicio, selecciona 'Sport'.

>>> Sports
Tema assignat correctament.
Tema preferit: Sports

Busques un llibre amb ilustracions?
  Respon 'Yes' si desitges un llibre amb ilustracions, o 'No' si
  prefereixes un llibre sense ilustracions.
```



```
>>> Yes
```

```
La teva resposta ha estat rebuda correctament:
Llibre amb il.lustracions: TRUE
```

```
Fin de la Ejecucion.
```

```
Title: Siesta en la Habitacion Nubosa | Rating: 9.89168531121605
```

```
Title: Safari por el Zoo Fabuloso | Rating: 9.77635748040592
```

```
Title: Cena en el Comedor Girasol | Rating: 9.67698707509204
```

7.4.4 Prototip 4 - Regles Específiques (Part 2)

Aquest nou prototip afegeix una sèrie de regles noves que amplien el coneixement que podrà gestionar l'SBC. Aquestes noves regles consisteixen en els trigger warnings, els atributs fixes i els subgèneres.

Implementació del Mòdul TRIGGER_W La integració del mòdul TRIGGER_W va sorgir com una necessitat crítica per abordar preguntes específiques relacionades amb el gènere literari seleccionat pel lector. La nostra observació inicial va ser que la pregunta “Vols evitar algun trigger en els llibres que llegeixes?” estava inextricablement lligada al gènere del llibre, donat que certes temàtiques sensitius estan més prevalentment associades amb gèneres específics.

Ens vam adonar que, encara que aquesta pregunta fos crucial, la seva aplicabilitat variava en funció del gènere del llibre i, per tant, no podia ser universal. A més, volíem evitar la duplicació de codi i garantir que el sistema fos tan eficient i escalable com fos possible. A continuació es presenten alguns dels desafiaments i solucions específiques que vam implementar.

Activació Basada en el Gènere Per garantir que aquesta pregunta es presentés de manera oportuna, vam implementar condicions específiques. En el cas dels adolescents, el mòdul TRIGGER_W s'activa només si els gèneres seleccionats són Romance o Mystery:

```
(if (or (eq ?genre "Romance")
        (eq ?genre "Mystery"))
    then
    (focus TRIGGER_W))
```

Per als adults, la lògica és similar, però s'expandix per incloure gèneres com Contemporary, Mystery, Romance i Horror:

```
(if (or (eq ?main_genre "Contemporary") (eq ?main_genre "Mystery")
        (eq ?main_genre "Romance") (eq ?main_genre "Horror"))
    then
    (focus TRIGGER_W))
```

Això assegura que el mòdul TRIGGER_W només s'activa quan és rellevant, millorant així l'eficiència del sistema i l'experiència de l'usuari.

Flexibilitat per a l'Usuari Vam voler donar als lectors l'oportunitat de personalitzar les seves preferències. Per això, vam introduir una regla que permet a l'usuari decidir si vol evitar triggers específics:

```
(defrule TRIGGER_W::pregunta-sobre-triggers
  (declare (salience 2))
  ?lector <- (object (is-a Lector) (triggerw ""))
  =>
  (if (yes-or-no-p "Vols evitar algun trigger en els llibres que llegeixes?"
                  (yes/no) ""))
```

Adaptabilitat i Millora Contínua Ens vam adonar que la naturalesa multivariant dels triggers warnings necessitava un enfocament més flexible. Per tant, vam reconfigurar l'atribut `triggerw` com un multislot, permetent als usuaris especificar múltiples triggers que volien evitar.

```
(multislot triggerw
  (type STRING)
  (create-accessor read-write))
```

A més, vam crear una funció personalitzada que permetia una selecció múltiple, oferint una major granularitat i control a l'usuari sobre les seves preferències.

```
(deffunction MAIN::multi-select-p (?question $?allowed-values)
  ...
  (return ?answers))
```

Atributs Fixes

La concepció dels atributs fixes és un altre aspecte clau en la nostra arquitectura de sistema d'experts. A través de l'anàlisi de la nostra base de dades, vam identificar atributs que, mentre són variables en alguns segments de públic, són constants en altres. Aquesta observació va ser crucial, ja que ens va permetre optimitzar el nostre sistema, assignant aquests atributs de manera predefinida depenent del públic, reduint així la necessitat de preguntes addicionals i millorant l'eficiència general del sistema.

Assignació d'Atributs Fixes per Públic Per implementar aquesta estratègia, vam crear regles específiques dins de cada mòdul de públic que assignen automàticament aquests atributs fixes. A continuació es descriuen els atributs fixes assignats per cada segment de públic:

- **BABY:** En el mòdul BABY, els llibres il·lustrats i les cobertes dures són comuns, i el temps de lectura és generalment curt. Per tant, vam assignar aquests atributs com a constants.

```
(defrule BABY::atributs
  ?lector <- (object (is-a Lector))
  =>
  (send ?lector put-illustrated TRUE)
  (send ?lector put-cover "hard")
  (send ?lector put-time_read "short"))
```

- **CHILD:** Per als nens, la coberta dura és un atribut constant, donada la seva durabilitat i adequació per a aquest grup d'edat.

```
(defrule CHILD::atributs
  ?lector <- (object (is-a Lector))
  =>
  (send ?lector put-cover "hard"))
```

- **TEEN:** En aquest cas, no vam identificar atributs que foren constants per a tots els adolescents, permetent una major flexibilitat i personalització.
- **ADULT:** Els adults generalment prefereixen llibres no il·lustrats, per la qual cosa aquest atribut es configura com a FALSE per defecte.

```
(defrule ADULT::atributs
  ?lector <- (object (is-a Lector))
  =>
  (send ?lector put-illustrated FALSE))
```

Impacte en la Eficiència i la Personalització La implementació d'atributs fixes segons el públic té múltiples avantatges. Per una banda, optimitza el procés de recomanació, eliminant la necessitat de consultar a l'usuari sobre preferències que ja es poden deduir del seu segment de públic.

Regla per Subgèneres

Per a l'obtenció de informació més concreta, hem afegit una regla per a la selecció de subgèneres literaris, així complementant la pregunta de gènere principal. Aquesta pregunta ofereix opcions més

específiques que reflecteixen les preferències del lector.

Selecció Dinàmica de Subgènere La funció `subg-disp` s'ha creat per mostrar dinàmicament les opcions de subgènere disponibles basades en el gènere principal seleccionat anteriorment pel lector. Aquesta funció recull tots els subgèneres associats amb un gènere específic i els presenta a l'usuari per a la seva selecció.

```
(deffunction ADULT::subg-disp (?genre_name)
  ; Inicializar la lista de subgéneros elegidos
  (bind ?subgenres_chosen (create$))

  (bind ?subgenres (find-all-instances ((?s Subgenero)) TRUE))

  (foreach ?s ?subgenres
    (bind ?sname (send ?s get-name_genre))
    (bind ?g (send ?s get-subgenre_of))
    (bind ?gname (send ?g get-name_genre))

    (if (eq ?gname ?genre_name)
      then
        (bind ?sname (sym-cat ?sname))
        (printout t " " ?sname crlf)
        (bind ?subgenres_chosen (insert$ ?subgenres_chosen (+
          (length$ ?subgenres_chosen) 1) ?sname))
      )
    )
  )
)
```

Interacció amb l'usuari Amb la llista de subgèneres obtinguda, la regla `ADULT::subgenere` s'activa, guiant a l'usuari a través del procés de selecció. Aquesta regla s'encarrega de realitzar la pregunta i registrar la selecció de l'usuari, associant el subgènere escollit amb la instància del lector.

```
(defrule ADULT::subgenere
  (declare (salience 3))
  ?lector <- (object (is-a Lector)
    (main_genre ?main_genre&:(neq ?main_genre [nil])))

  =>
  (bind ?genre_name (send ?main_genre get-name_genre))
  (printout t "A partir de l'eleccio del genere, pots escollir quin subgenere
    d'aquest t'agradaria." crlf)
  (printout t "Quin subgenere t'agradaria? Pel genere escollit (" ?genre_name ")
    tens les següents opcions:" crlf)
  (bind ?subgeneres (subg-disp ?genre_name))

  (bind ?sub_genre (basicQ "
  >>> " ?subgeneres))
  (bind ?sub_genre (str-cat ?sub_genre))
  (bind ?sub_genrei (find-instance
    ((?g Subgenero))
    (eq (send ?g get-name_genre) ?sub_genre)
  ))
  (send ?lector put-has_subgenre ?sub_genrei)
)
```

Adaptabilitat i Personalització A través de la implementació d'aquest sistema de subgèneres, hem augmentat significativament la capacitat del nostre SBC per adaptar-se a les preferències individuals dels usuaris. La integració de subgèneres no només millora la precisió de les recomanacions, sinó que

també enriqueix l'experiència del lector, oferint títols que s'alineen més estretament amb els seus gustos i interessos particulars.

7.4.5 Prototip 5 - Question skip, Sistema de Puntuació i Poda per TriggerWarnings

En aquesta fase del desenvolupament, l'objectiu principal va ser millorar l'experiència de l'usuari i preparar el terreny per a una selecció més refinada de llibres recomanats. Aquesta etapa va introduir una funcionalitat que permet a l'usuari ometre preguntes i també va establir les bases per a un sistema de puntuació per la selecció del llibre a recomanar.

Implementació del Skip La llibertat i la comoditat de l'usuari són d'una importància capital. Així, la capacitat de permetre als usuaris saltar-se preguntes que prefereixen no respondre, o aquelles en les quals poden estar indecisos, va ser identificada com una característica crucial. Aquesta funcionalitat garanteix que els usuaris no se sentin coartats ni forçats a proporcionar informació que podria ser sensible o incòmoda per a ells, o que simplement prefereixen mantenir en privadesa. A més, permet que evitin influenciar involuntàriament les recomanacions finals amb informació que podria no reflectir les seves preferències reals o situacions específiques de lectura. Aquesta adaptabilitat en la recopilació de dades reforça la personalització i el respecte cap a la individualitat de cada usuari.

La implementació d'aquesta funcionalitat va requerir una revisió de la funció `basicQ` per incloure el caràcter "-" com una resposta acceptada. Això indica que l'usuari ha escollit ometre la pregunta en qüestió.

```
(bind ?options (insert$ ?options (+ (length$ ?options) 1) -))
```

Aquest mateix principi va ser aplicat a la funció `yes-or-no-p`, assegurant-se que la llibertat de l'usuari per saltar-se preguntes fos consistent a través de tot el sistema.

```
(bind ?response (ask-question ?question yes no y n YES NO Yes No -))
```

Per el multi-select-p., que es va incloure un condicional més.

```
(if (member$ ?item ?allowed-values) and (neq ?item "-"))
```

Per gestionar aquesta nova opció de resposta, es va implementar una condició addicional per evitar que el caràcter "-" s'introduís com una dada d'entrada en el perfil de l'usuari. Això assegura que les omissions no afectin la qualitat o l'exactitud de les recomanacions.

```
(if (eq ?time_read "-")
  then
    (printout t "Has decidit saltar-te aquesta pregunta." crlf)
  else
    (send ?lector put-time_read ?time_read)
    (printout t "Llargaria del llibre: " ?time_read crlf)
)
```

Aquest enfocament millora la flexibilitat del sistema, permetent una interacció més adaptativa i respectuosa amb l'usuari.

Implementació del Sistema de Puntuació L'objectiu principal d'introduir un sistema de puntuació al nostre SBC de recomanació de llibres és afinar la selecció de les recomanacions i proporcionar al lector llibres que s'ajustin el màxim possible a les seves preferències i interessos.

Cada llibre de la nostra base de dades comença amb una puntuació inicial establerta per l'atribut *rating*. Hem definit una regla específica per a cada atribut d'aquests llibres. Aquestes regles tenen la funció de recórrer tots els llibres disponibles i comprovar la compatibilitat amb les respostes del lector. Si un llibre es considera compatible amb l'atribut específic de la resposta del lector, aquest llibre es bonifica amb 10 punts. Això significa que si un llibre és totalment compatible amb totes les respostes del lector, obtindrà la puntuació màxima possible segons el nombre d'atributs que s'han considerat i segons el seu rating.

Una vegada s'han aplicat totes les regles i s'han assignat les puntuacions a cada llibre, utilitzem la funció `bestbooks` per extreure els llibres amb les puntuacions més altes. Amb `bestbooks ?N`, obtenim els N llibres amb la puntuació més alta, sent N el nombre de llibres que volem recomanar. Aquesta selecció assegura que l'usuari rebi recomanacions basades en la seva compatibilitat màxima amb els llibres disponibles a la base de dades. Imprimeix el llibre i dues característiques en el format:

```
"Title: " <Títol del llibre> " | Puntuacion: " <Puntuació final del llibre>
```

Amb aquest sistema de puntuació, hem aconseguit oferir recomanacions més refinades i ajustades a les preferències individuals dels lectors, augmentant així la probabilitat que els llibres suggerits siguin del seu interès i s'ajustin a les seves expectatives.

Implementació de la Poda per Trigger Warnings A causa de la importància de l'atribut *Trigger Warnings*, hem arribat a la conclusió que simplement restar punts als llibres que contenen continguts delicats que l'usuari no vol trobar i sumar punts als altres llibres no és suficient per garantir una experiència satisfactòria.

Creiem que quan un usuari sol·licita evitar un contingut específic, és de vital importància garantir que no se li presenti de cap manera. La simple penalització de la puntuació podria no ser prou efectiva a la pràctica, ja que alguns continguts delicats podrien passar desapercebuts o no rebre l'atenció necessària. Per aquesta raó, hem optat per dur a terme una poda més enèrgica, consistent en l'eliminació de tots els llibres que tracten temes delicats que el lector no vol llegir.

Aquesta poda s'executa mitjançant una regla específica que es troba dins del mòdul `TRIGGER.W`.

```
(defrule TRIGGER_W::podaTriggers
  (declare (salience 4))
  ?lector <- (object (is-a Lector)
                    (public ?p))

  =>
  (bind ?llibres (find-all-instances ((?l Libro)) (eq ?l:public ?p)))
  (bind ?trig (send ?lector get-triggerw))
  (printout t ?trig crlf)
  (progn$ (?l ?llibres)
    (bind ?tw (send ?l get-triggerw))
    (bind ?comptador FALSE)
    (progn$ (?t ?tw)
      (if (member$ ?t ?trig) then
        (bind ?comptador TRUE)))
    (if (eq ?comptador TRUE) then (send ?l delete))
  )
)
```

7.4.6 Prototip Final

El nostre prototip final ha incorporat diverses millores i afegits respecte a les versions anteriors, amb l'objectiu d'optimitzar la interacció amb l'usuari i proporcionar recomanacions de llibres de més alta qualitat.

Interacció amb l'Usuari: Hem treballat en la millora del text de sortida perquè sigui més proper i amigable amb l'usuari. Ara, el sistema es comunica d'una manera que fa que el lector es senti còmode i guiat durant tot el procés de recomanació.

A continuació es mostra el text de sortida del començament:

CLIPS> (run)

```

-----
/          |/          |   Recomanador de Llibres
||         |          ||
||         |          ||   Practica 1 - SBC
||         |          ||
||         |          ||   By:
||         |          ||   Artur A A
||         |          ||   Llum F P
||/        |/         |||   Paula G S
                        |||   Oscar M S
-----

                BENVINGUT!

```

Li farem diverses preguntes i el nostre sistema recomanador li oferirà els llibres que més s'adapten a les seves preferències.

A quin grup d'edat pertanyes?

```

Si tens entre 4 i 6 anys tria l'opcio 'baby'
Si tens entre 7 i 14 anys tria l'opcio 'children'
Si tens entre 15 i 18 anys tria l'opcio 'teenager'
Si tens mes de 18 anys tria l'opcio 'adult'

```

Qualitat de la Recomanació: Hem desenvolupat un sistema de puntuació que avalua quan representatiu és el resultat en funció del que busca el client:

- *"Què bé! Aquest llibre és exactament el que estaves buscant"*: Aquesta resposta indica que el llibre recomanat coincideix plenament o casi plenament amb les especificacions que el lector ha proporcionat.
- *"Aquest llibre pot ser una bona opció"*: En aquest cas, el llibre recomanat comparteix algunes, però no totes, de les característiques desitjades pel lector.
- *"Hmm... Mira veure si t'agrada aquest"*: Si el sistema retorna aquest missatge, significa que no hem pogut trobar un llibre en la nostra base de dades que compleixi estrictament amb les especificacions del lector.

Cal destacar que si un usuari decideix ometre algunes preguntes, això pot afectar la qualitat de la recomanació. Aquesta baixada de qualitat es reflecteix en el missatge que es mostra a l'usuari. Aquesta interacció s'ha fet amb el sistema de puntuació.

```

(deffunction JUSTIFY::getRecommendationRange (?lectorType ?puntuacio)
  (if (eq ?lectorType "baby")
    then
      (if (>= ?puntuacio 40)
        then (return "ideal")
        else (if (and (>= ?puntuacio 30) (< ?puntuacio 40))
          then (return "good")
          else (return "bad"))))

  else if (eq ?lectorType "children")

```

```

    then
      (if (>= ?puntuacio 35)
        then (return "ideal")
        else (if (and (>= ?puntuacio 20) (< ?puntuacio 35))
          then (return "good")
          else (return "bad")))
    else if (eq ?lectorType "teen")
      then
        (if (>= ?puntuacio 45)
          then (return "ideal")
          else (if (and (>= ?puntuacio 10) (< ?puntuacio 45))
            then (return "good")
            else (return "bad")))
    else if (eq ?lectorType "adult")
      then
        (if (>= ?puntuacio 85)
          then (return "ideal")
          else (if (and (>= ?puntuacio 10) (< ?puntuacio 85))
            then (return "good")
            else (return "bad")))
  )
)

```

Sistema de Justificació dels Resultats: Per proporcionar transparència en el nostre procés de recomanació, hem implementat un sistema de justificació. Aquest sistema compara les especificacions proporcionades pel client amb les del llibre recomanat, indicant quines característiques tenen en comú. D'aquesta manera, l'usuari pot entendre per què s'ha seleccionat un llibre en particular per a ell.

```

(defrule JUSTIFY::justify
  ?lector <- (object (is-a Lector)
    (actual ?isactual)
    (bestseller ?isbestseller)
    (cover ?coverType)
    (goal ?mainGoal)
    (illustrated ?isIllustrated)
    (time_read ?timeread)
    (theme ?mainTheme)
    (public ?public)
    (main_genre ?genereMain)
    (has_subgenre ?subgenere))

  =>
  (bind ?llibres_finals (bestbooks 3))
  (foreach ?book ?llibres_finals
    (showProperties ?book ?lector)
    ;actual
    (if (and (neq ?isactual nil)(eq ?isactual (send ?book get-actual)))
      then (if (eq ?isactual TRUE)
        then (printout t "Aquest llibre es actual tal i com has demanat." crlf)
        else (printout t "Aquest llibre ha estat escrit per un gran autor classic,
          tal i com has demanat." crlf)))
    ;bestseller
    (if (and (neq ?isbestseller nil)(eq ?isbestseller (send ?book get-bestseller)))
      then (if (eq ?isbestseller TRUE)
        then (printout t "Aquest llibre es un bestseller tal i com has demanat." crlf)))
    ;cover
    (if (and (and (neq ?coverType "")(eq ?coverType (send ?book get-cover)))
      (and (neq ?public "child")(neq ?public "baby"))))

```

```

    then (switch ?coverType
      (case "hard" then (printout t "Hem trobat un llibre de tapa dura tal i com
                                   volies." crlf))
      (case "soft" then (printout t "Hem trobat un llibre amb tapa fina tal i com
                                   volies." crlf))
      (default (printout t "Hem trobat un llibre de butxaca perquè el puguis portar a
                           qualsevol lloc! Tal com volies." crlf)) ))

;goal
  (if (and (neq ?mainGoal "") (eq ?mainGoal (send ?book get-goal))) )
    then (if (eq ?mainGoal "Entertainment")
      then (printout t "Segur que amb aquest llibre passes una bona estona entretenint-te
                       amb la lectura!" crlf)
      else (printout t "Segur que si llegeixes aquest llibre aprens moltes coses
                       interessants!" crlf)))

;illustrated
  (if (and (neq ?isIllustrated nil)
    (and (eq ?isIllustrated (send ?book get-illustrated)) (eq ?public "children"))))
    then (if (eq ?isbestseller TRUE)
      then (printout t "Aquest llibre conte dibuixos i il·lustracions, tal i com has
                       demanat." crlf)
      else (printout t "Aquest llibre no conte dibuixos, només text, tal i com has
                       demanat." crlf ))

;theme
  (if (and (neq ?mainTheme "") (eq ?mainTheme (send ?book get-theme))) )
    then (tematica ?mainTheme))

;genre
  (bind ?g (send ?book get-main_genre))
  (if (neq ?genereMain [nil])
    then
      (bind ?g_name (send ?g get-name_genre))
      (bind ?gen_name (send ?genereMain get-name_genre))
      (if (eq ?g_name ?gen_name)
        then (genre ?g_name) ))

;subgenre
  (bind ?s (send ?book get-has_subgenre))
  (if (neq ?subgenere [nil])
    then
      (bind ?s_name (send ?s get-name_genre))
      (bind ?sub_name (send ?subgenere get-name_genre))

      (if (eq ?s_name ?sub_name)
        then (printout t "Com haves demanat, el llibre es del subgenere " ?s_name crlf) ))
    (printout t crlf)
  )
)

```


8 Jocs de prova

Els jocs de prova són crucials per a demostrar les capacitats del nostre sistema basat en el coneixement. Hem creat dos conjunts de jocs de prova per il·lustrar les habilitats del sistema de manera concreta i efectiva. Això ajuda a validar la funcionalitat del sistema i a mostrar la seva capacitat per abordar una àmplia gamma de situacions.

8.1 Jocs de prova a partir d'instàncies

L'arxiu `generador_jp_instancias.py` defineix la generació d'un arxiu de Clips amb un gran nombre d'instàncies de lectors. En executar aquest arxiu, es requereixen cinc arguments que s'han de passar mitjançant `sys.argv` com segueix:

`sys.argv[1]`: Quantitat de lectors del grup d'edat 'baby'.
`sys.argv[2]`: Quantitat de lectors del grup d'edat 'children'.
`sys.argv[3]`: Quantitat de lectors del grup d'edat 'teenager'.
`sys.argv[4]`: Quantitat de lectors del grup d'edat 'adult'.
`sys.argv[5]`: Nom que vulguis assignar a l'arxiu '.clp' resultant.

A partir d'aquí, es generen instàncies dels lectors en la quantitat especificada per a cada grup d'edat, i es crea un arxiu de sortida amb l'extensió `.clp` que conté aquestes instàncies. Això permet generar una gran quantitat de dades d'instàncies de lectors en funció de les diferents categories d'edat especificades, tal i com es mostra a continuació:

```
(definstances instances
  ([lector] of Lector
    (public "baby")
    (goal "Learning")
    (hab_lectura "Neutral")
    (illustrated TRUE)
    (theme "Sports")
  ))

(definstances instances
  ([lector] of Lector
    (public "children")
    (hab_lectura "Neutral")
    (illustrated TRUE)
    (theme "Daily_life")
    (time_read "short")
  ))

(definstances instances
  ([lector] of Lector
    (public "teenager")
    (main_genre [Dystopia])
    (bestseller TRUE)
    (cover "soft")
    (hab_lectura "Neutral")
    (time_read "short")
  ))

(definstances instances
  ([lector] of Lector
    (public "adult")
    (main_genre [Fantasy])
```

```
(has_subgenre [Urban_Fantasy])
(actual FALSE)
(famous FALSE)
(bestseller FALSE)
(cover "soft")
(hab_lectura "Neutral")
(time_read "medium")
))
```

Tot i que genera un arxiu en format CLIPS, aquest no es pot executar de cop, ja que el nostre sistema està pensat per a que únicament existeixi un únic lector. SI carreguessim aquesta pàgina tindriem més lectors i no funcionaria correctament. Si es vol generar el fitxer resultant d'una sola instància és suficient amb demanar 1 de un public concret i 0 dels altres. D'aquesta manera es pot obtenir un fitxer *joc_de_prova.clp* que es pot carregar juntament amb el nostre fitxer *main_sin_instancias.clp*, que es una còpia del main però sense la instància de lector buida.

Tenim un exemple de la generació a la carpeta */jocs_prova/joc_prova_instancias*. L'arxiu *joc_prova.clp* és un exemple de joc de prova en forma de instància i *joc_prova_out.txt* és el resultat d'executar la línia de comandes:

```
cd .\jocs_prova\
\jocs_prova> python ejecutador.py joc_prova_instancias/jp_comandes1 joc_prova_instancias/joc_prova.clp
```

8.2 Jocs de prova amb executador

L'etapa de proves és fonamental en qualsevol sistema de recomanació, ja que ens permet assegurar-nos que les recomanacions generades són les esperades i que la lògica interna funciona de manera òptima. Aquesta subsecció descriu la manera en què hem creat i executat diferents jocs de prova amb un executador automàtic.

8.2.1 Generació de Comandaments

Per a la generació de jocs de prova, es va utilitzar la funció `generar_libros_aleatorios` implementada dins l'arxiu *generador_jp_terminal.py*, que permet generar respostes aleatòries en base a diferents perfils d'usuaris: baby, children, teenager i adult. Aquesta funció crea un conjunt de respostes que simula les preferències de lectura d'aquests usuaris i les desa en el fitxer *jocs_comandes.txt*. Es compleixen totes les restriccions i es segueix la salience per saber quina pregunta va abans o després.

Una vegada generats aquests comandos, la funció principal del script guarda les respostes en un fitxer de text amb l'estructura següent:

```
(load ../main.clp)
(load ../instances.clp)
... respostes generades ...
(exit)
```

L'estructura d'aquests comandaments garanteix que, al ser executats, el sistema carregarà les regles i les instàncies necessàries, processarà les respostes generades i, finalment, tancarà la sessió de CLIPS.

8.2.2 Execució Automàtica de Jocs de Prova

Una vegada generats els comandaments, cal executar-los dins l'entorn de CLIPS per observar i analitzar els resultats. Per a això, es va crear un script d'execució automàtica. Aquest script, anomenat *ejecutador.py*, utilitza la funció `run_clips_with_commands` per iniciar un procés amb l'entorn de CLIPS i, posteriorment, llegeix i executa els comandos desats en el fitxer generat anteriorment.

La sortida generada per CLIPS, que inclou les recomanacions i altres missatges, es captura i es desa en un fitxer de text especificat. Aquesta sortida pot ser utilitzada després per analitzar la qualitat de les recomanacions, detectar possibles errors en les regles o millorar la lògica del sistema.

8.2.3 Exemple d'ús

Per utilitzar els jocs de prova cal fer les següents passes: Passa 1: generar el joc de proves en format txt en el fitxer `../jocs_prova/jocs_comandes.txt`

```
generator> python generador_jp_terminal.py 1 2 1 2
```

Quan ja tenim el document de text amb les comandes relatives a CLIPS, ja podem utilitzar l'executador per llançar el joc de proves i obtenir els resultats. Ho fem amb la comanda següent:

```
jocs_prova> python ejecutador.py jocs_comandes jocs_output
```

8.2.4 Jocs de prova manuals

Aquest generador té unes limitacions i és que no hem pogut modelar l'hàbit de lectura ja que creava interferències entre classes de *public*. Tampoc no hem afegit el guió per a saltar la pregunta ja que no ens interessa en les nostres execucions de prova. Així doncs, hem fet un joc de proves manual per a mirar el seu funcionament. Aquest és *manual_comandes.txt* i es pot executar amb el mateix *ejecutador.py*.

8.3 Comparació dels dos mètodes

El mètode de creació de la instància pot ser útil per poder gestionar casos en que es tinguin els perfils dels lectors definits i no es vulgui realitzar una interacció. En aquest cas, ens estariem saltant el subproblema de l'imput obtenint el perfil del lector. Seria una tècnica adient si tenim bases de dades de lectors. En el resultat no es veu la interacció de preguntes ja que no entrem en cap regla de pregunta degut a que la instància de lector ja està omplerta.

D'altra banda, el mètode de creació de comandes sí que mostra la suposada interacció entre el lector i el recomanador. En aquest cas, podem veure totes les interaccions que es produeixen entre el sistema i l'usuari, representades per les preguntes i respostes que es van succeint. Aquesta tècnica resulta especialment útil quan volem simular un escenari real d'interacció amb l'usuari o quan desitgem testar i validar les preguntes que fa el sistema i com aquest respon segons les respostes proporcionades. A més, ens permet executar d'una forma semi-automàtica execucions amb skip de preguntes.

En resum, mentre que el mètode basat en instàncies és més ràpid i directe, el mètode basat en comandes proporciona una visió més completa de la interacció i pot ser més adequat per a situacions en què la simulació de la interacció real sigui una prioritat. La elecció entre un mètode o un altre depèn de les necessitats específiques del projecte i dels objectius de les proves que es vulguin realitzar. En el nostre cas és més útil la generació de comandes i output ja que és del nostre interès visualitzar les interaccions.

9 Futures Ampliacions

Durant el procés d'elaboració d'aquesta pràctica, van sorgir diverses idees d'implementació per al sistema. Tot i que vam identificar moltes opcions interessants, les limitacions de temps ens van portar a centrar-nos en les que consideràvem més rellevants. No obstant això, encara disposem d'una sèrie d'ampliacions conceptualitzades que podrien ser implementades en un futur per enriquir i fer més complet el nostre Sistema Basat en el Coneixement. Aquestes ampliacions podrien millorar encara més la funcionalitat i el realisme del sistema en el futur.

9.1 Atribut dificultat

Com es pot veure en els apartats de formalització i implementació, aquest atribut ja està inclòs com a part de la classe "Llibre". La idea era avaluar la quantitat de paraules diferents en un llibre, com a indicador de la complexitat o riquesa del seu vocabulari.

Tenint en compte que la vostra base de dades ha estat generada de manera artificial, l'assignació de valors per a la quantitat de vocabulari hauria de tenir en compte factors com el públic al que està associat el llibre i la seva longitud. Aquesta metodologia pot ser efectiva per a proporcionar una mesura més precisa de la complexitat lingüística dels llibres.

La conversió d'aquest atribut numèric en una variable categòrica amb els valors "difícil," "mitja" i "fàcil," en relació amb la quantitat de paraules diferents en un llibre, facilitaria la distribució i l'ús d'aquesta informació. A més, permetria incorporar-lo com una pregunta amb opcions de resposta tancada, seguint l'estructura del nostre sistema de regles.

És important destacar que, abans de la implementació d'aquest atribut, hauríem de realitzar una recerca d'informació experta per a assegurar una base sòlida i precisa per a la seva construcció.

9.1.1 Habít de lectura-adults

Actualment, l'atribut "hàbit de lectura" ja s'aplica, però per el perfil de "adults" no aporta informació en el procés de recomanació de llibres. Per tant, seria necessari una revisió d'aquesta qüestió per a fer-la més significativa. Una opció seria vincular-la amb l'atribut de dificultat, on els hàbits de lectura puguin indicar la preferència per textos més complexos o més simples. Així, aquesta informació esdevindria més rellevant, aportant dades clau per a afinar millor les recomanacions de llibres segons el nivell de complexitat que cada adult prefereix o està preparat per abordar.

9.2 Atribut Sagas

De manera similar a l'atribut "dificultat," aquest atribut també ja està incorporat com a part de la classe "Llibre." La nostra idea era crear col·leccions per als públics més joves (com a bebès i nens) i sagas per als públics més grans (adolescents i adults) per a millor organitzar i oferir llibres específics per a cada grup d'edat.

Inclouríem per tant una nova capa de classificació per a identificar si un llibre pertany a una saga, i d'aquesta manera, oferir opcions que fomentin la continuïtat en la lectura. Per exemple, si un usuari mostra interès en un llibre d'una saga, el sistema podria recomanar altres llibres de la mateixa sèrie. Així mateix, aquest atribut ajudaria a evitar recomanacions de llibres que siguin parts avançades de sagas que l'usuari no ha començat.

9.3 Atribut Idiomes

L'atribut d'idiomes es planteja com una pregunta universal, dirigida a tots els usuaris independentment de la seva edat. Això garantiria que les preferències lingüístiques siguin sempre considerades en el procés de recomanació de llibres. La inclusió d'aquesta pregunta permetria adaptar les suggerències no només al gust literari, sinó també a l'idioma en què l'usuari se sent més còmode llegint o en el qual desitja llegir, ja sigui per gust personal, per pràctica lingüística o per altres raons educatives.

9.4 Mètode de puntuació més complex

Es planteja un sistema de puntuació més sofisticat que assigni pesos diferents a les preguntes en funció de la seva rellevància o capacitat d'aportar informació significativa sobre les preferències de l'usuari. Això podria incloure l'ús d'algoritmes que aprenguin de les interaccions passades per a afinar millor les recomanacions futures, oferint així suggeriments més precisos i personalitzats.

També podríem fer, per exemple, una suma ponderada segons la rellevància. En aquest cas es podria modelar un nou atribut que fos si a la persona li interessen els ratings d'altres persones ja que potser no li interessen les valoracions d'altres lectors. En la implementació actual el rating sempre compta per al càlcul de la puntuació.

9.5 Mètode de justificació més complex

Podríem ampliar la justificació de l'elecció de llibres. En la nostra implementació només donem informació si coincideixen les característiques del llibre. Podria ser bona idea, donar un missatge explicant que no hem trobat cap llibre amb aquella característica demanada, però que li pot agradar el llibre; així podríem advertir al lector sobre aquelles característiques que no es compleixen. També seria interessant poder dir al lector que

9.6 Historial de llibres

Incorporar un sistema que tingui en compte l'historial de lectures dels usuaris podria significar un gran avanç en la personalització de les recomanacions. Això permetria identificar patrons en les preferències de lectura, com ara gèneres favorits, autors recurrents o tipus de narratives. Amb aquesta informació, el sistema podria predir amb major precisió els interessos actuals i futurs de l'usuari, millorant així la qualitat de les recomanacions ofertes.

10 Conclusions

En aquesta pràctica hem desenvolupat un sistema expert recomanador de llibres que deixa a l'usuari introduir les seves preferències per tal d'analitzar-les i, mitjançant un procés de selecció (en el nostre cas un sistema de puntuació), recomanar aquells llibres que més s'hi adapten.

Durant tot el conjunt de la pràctica ens hem trobat amb moltes dificultats a resoldre. L'obtenció de dades i fonts fiables de distribucions probabilístiques ha estat tot un repte, tant per l'absència d'elles com per la no-legitimitat o manca d'ètica de la seva procedència.

També ho ha sigut la part de justificar les decisions del nostre SBC. Un bon sistema basat en coneixement ha de ser capaç de justificar les seves decisions, de fet és una de les propietats que li donen potencial, rigidesa i explicabilitat. Per aquest motiu ens vam veure amb l'obligació d'ampliar el nostre sistema i crear mòduls extres per aquesta tasca que no teniem en ment a l'inici.

Aquest és un molt bon exemple de què és un procés iteratiu (metodologia incremental). I és que durant el transcurs de la pràctica hem anat millorant i afegint característiques al nostre sistema.

Com hem mencionat a l'apartat anterior, ens han quedat moltes millores en ment com la de relacionar autors per similitud, com la d'afegir sagues, sobretot pel públic jove i nen o com la d'afegir diversos idiomes.

Tot i així estem molt contents amb el resultat, la col·laboració de tot l'equip ha estat clau per encarar la pràctica i poder dur-la a terme. Tots els companys creiem haver aportat molt de nosaltres i finalment hem pogut desenvolupar un sistema expert força robust amb totes les seves parts.

11 Appèndix 1

Pregunta 1: Què és per a tu la literatura?

Com a concepte és una narració d'històries. Per a mi ha de tenir alguna cosa artística. No ha de ser un diari, ha de tenir art.

Pregunta 2: Quin tipus de lectora ets?

Obsessiva, devoradora, llegeixo de tot, constant.

En aquest apartat volem obtenir informació sobre les diferents consideracions que cal tenir en compte en la interacció amb el lector.

Pregunta 3: Quines són les preguntes indispensables que has de fer independentment del perfil del lector?

- Edat, que es divideix en:
 - 4-6 anys, molt joves.
 - 7-14 anys, nens.
 - 15-18 anys, adolescents.
 - Més de 18 anys, adults.
- Si el llibre és Realista o Fictici
- Idioma en que ha estat escrit

Pregunta 4: Si haguessis de discriminar la població de lectors en diferents tipus, quants en serien i quins?

Els dividiria en 15 tipus de lectors diferents, que serien:

Gran lector (de moltes fonts, tot i que tots amb preferència), misteri, fantasia, romàntica, realista (contemporànies), poesia, interactives (proposen reptes), informatius (biòpics), historiadors, d'autoajuda, espiritisme, lector ocasional, lectors d'influencers, de còmic.

Pregunta 5: Quines altres preguntes acostumes a fer segons el perfil?

- Si classifiquem als lectors segons l'EDAT:
 - 4-6 anys:
 - * Què t'agrada? (els pares realment els és indiferent el llibre)
 - 7-14 anys:
 - * Tens molta costum de lectura? [si té 13 o més i té costum de lectura, passa a adolescent]
 - * Quin és el teu hobby?
 - * Vols algun llibre d'un personatge en concret? Quin?
 - * Has llegit llibres d'alguna col·lecció o saga?
 - * Busques un llibre amb il·lustracions?
 - 15-18 anys:
 - * Tens molta costum de lectura? [si té 16 o més i té costum de lectura, passa a adult]
 - * Quina obra t'ha agradat abans?

- * T'agraden els llibres de moda entre les últimes novetats?
- * Vols tapa dura, tapa tova o edició de butxaca?
- * Vols evitar algun trigger warning? [violació, assassinats brutals i descriptius...]
- Més de 18 anys:
 - * Quin gènere estàs buscant?
 - * Qui és el teu autor preferit?
 - * Vols algun autor reconegut?
 - * Quan temps tens per llegir?
 - * T'agraden les novel·les actuals o clàssiques?
 - * T'interessa llegir best-sellers? (sí, no, indiferent)
 - * Estàs buscant una edició especial d'un llibre?
 - * Vols tapa dura, tapa tova o edició de butxaca?
- Si classifiquem als lectors segons el GÈNERE que llegeixen:
 - Contemporània:
 - * T'agradaria una novel·la històrica?
 - * Prefereixes un llibre d'època o filosòfic?
 - Misteri:
 - * Quin tipus de misteri prefereixes? [selecció]
 - * Vols evitar algun trigger warning? [violació, assassinats brutals i descriptius...]
 - Romància:
 - * Vols evitar algun trigger warning? [violació, assassinats brutals i descriptius...]
 - Contemporània:
 - * Vols evitar algun trigger warning? [violació, assassinats brutals i descriptius...]
 - * Autor nacional, català o estranger?
 - Terror:
 - * Vols evitar algun trigger warning? [violació, assassinats brutals i descriptius...]
 - Històrica-Realista:
 - * D'Espanya, Catalunya o estrangera?
 - * Pertany a un esdeveniment important?

En aquesta secció tenim la intenció d'informar-nos sobre com crear un arbre de gèneres i subgèneres real i lògic.

Pregunta 6: Quins són els gèneres principals en les novel·les de ficció?

Contemporània, romàntica, històrica, misteri, fantasia, terror i ciència-ficció.

Pregunta 7: Poden els subgèneres pertànyer a diversos gèneres diferents?

Sí, això fa que la categorització dels llibres sigui difícil i, al final, a les llibreries es categoritzen segons l'emoció principal que transmeten. Tot i així, hi ha ambigüitats i llibres que es mouen d'una secció a una altra.

Pregunta 8: Quins són els subgèneres de cadascun dels gèneres? Podries crear un arbre? [No és un arbre, és un graf, simplifiquem per a la comprensió de l'entrevistada]

Clicant sobre l'ARBRE arribareu a la representació de gèneres i subgèneres que ha dibuixat la Laura.

Cal destacar que els venedors utilitzen un arbre més reduït i, en canvi, la base de dades de la llibreria conté més subgèneres i informació addicional. L'arbre creat combina la informació de les dues fonts.

Pregunta 10: Quines característiques pot tenir un llibre?

Gènere, preu, format (butxaca, tapa dura, tapa tova), trigger warnings, realista/fictícia, il·lustrat o no, saga/col·lecció (acabada o no), idioma, nombre de pàgines, avaluació general (contemporània actualitat), tema i best seller.

Pregunta 11: Quines característiques té l'autor?

Si és clàssic o actual, el gènere principal que escriu, si és reconegut, i autors similars a ell.

Pregunta 12: Quins patrons segueixen els llibres del mateix autor? *Exemple: Un autor que escriu novel·les molt complicades pot escriure també llibres per a nens?*

No han de seguir sempre el mateix tipus de llibres. Normalment es solen cansar d'un gènere i poden escriure en dos gèneres diferents.

Pregunta 13: Vull comprar un llibre, em pots recomanar un?

Característiques del lector: 20 anys, li agraden llibres juvenils, realistes (descarta la fantasia), socials, de misteri, romàntics i d'aventura (sense màgia).

Si estigués a la botiga podria recomanar-te; de memòria em recomanaria un llibre que es diu "El brillo de las luciérnagas".