



Pràctica de Sistemes Basats en el Coneixement

Sistema Expert de Recomanació d'Habitatges de Lloguer

GRAU IA – Q1 CURS 2025-2026

Departament de Ciències de la Computació
Universitat Politècnica de Catalunya

ANEL ADEMOVIC

ALEIX PITARCH

`anel.suljic@estudiantat.upc.edu` `aleix.pitarch@estudiantat.upc.edu`

JOAN SOLINA

`joan.solina@estudiantat.upc.edu`

14 de desembre de 2025

Índex

1	Identificació del problema	7
1.1	Descripció del problema	7
1.2	Anàlisi de viabilitat	7
1.2.1	Viabilitat tècnica	7
1.2.2	Viabilitat de desenvolupament	8
1.2.3	Limitacions identificades	8
1.3	Fonts de coneixement	9
1.3.1	Fonts primàries	9
1.3.2	Fonts secundàries	9
1.4	Obtenció de la informació de l'usuari	9
1.4.1	Dades personals	9
1.4.2	Situació familiar i convivència	10
1.4.3	Pressupost	10
1.4.4	Ubicació i mobilitat	10
1.4.5	Accessibilitat	11
1.4.6	Mascotes	11
1.4.7	Preferències i exclusions	11
1.4.8	Generació del perfil	11
1.5	Objectius del sistema	11
1.5.1	Objectius funcionals	11
1.5.2	Objectius de qualitat	12
1.6	Resultats del sistema	12
1.6.1	Per a cada sol·licitant	12
1.6.2	Per a cada oferta recomanada	13
1.6.3	Informació complementària	13
2	Conceptualització	14
2.1	Conceptes del domini	14
2.1.1	Jerarquia de sol·licitants	14
2.1.2	Habitatges i les seves característiques	15
2.1.3	Serveis urbans	16
2.1.4	Localització	17

2.1.5	Oferta	17
2.1.6	Conceptes auxiliars	18
2.2	Descomposició en problemes i subproblemes	18
2.2.1	Fase 1: Inicialització	18
2.2.2	Fase 2: Abstracció	19
2.2.3	Fase 3: Descart	21
2.2.4	Fase 4: Puntuació (Scoring)	22
2.2.5	Fase 5: Classificació	25
2.2.6	Fase 6: Presentació	25
2.3	Exemples de coneixement expert	26
2.3.1	Context de l'elicitació	26
2.3.2	Exemple 1: Gestió del pressupost màxim	26
2.3.3	Exemple 2: Mida mínima viable de l'habitatge	27
2.3.4	Exemple 3: Famílies amb fills petits	27
2.3.5	Exemple 4: Grups d'estudiants	28
2.3.6	Exemple 5: Persones grans	29
2.3.7	Exemple 6: Criteris de classificació de recomanacions	30
2.4	Descripció informal del procés de resolució	30
2.4.1	Escenari d'ús típic	30
2.4.2	Flux general del sistema	34
3	Formalització	36
3.1	Introducció	36
3.2	Procés de Construcció de l'Ontologia	36
3.2.1	Metodologia Aplicada	36
3.3	Descripció de l'Ontologia	36
3.3.1	Visió General	36
3.3.2	Jerarquia de Classes	37
3.4	Relacions entre Conceptes	44
3.5	Descripció dels atributs	45
3.5.1	Atributs de Solicitant	45
3.5.2	Atributs d'Habitatge	45
3.5.3	Atributs de Servei	46
3.5.4	Atributs de Localització	46
3.5.5	Atributs d'Oferta	47

3.6	Conclusions	47
4	Implementació	48
4.1	Representació de l'ontologia	48
4.1.1	Traducció automàtica amb owl2clips	48
4.1.2	Estructura de les classes	48
4.1.3	Representació d'habitatges i serveis	49
4.1.4	Relacions entre objectes	50
4.1.5	Templates per a fets temporals	51
4.2	Modularització del sistema	52
4.2.1	Control de flux amb fases	52
4.2.2	Fase d'inicialització	52
4.2.3	Fase d'abstracció	54
4.2.4	Fase de descart	55
4.2.5	Fase de puntuació	56
4.2.6	Fase de classificació	57
4.2.7	Fase de presentació	57
4.3	Implementació de la metodologia de classificació heurística	58
4.3.1	Abstracció	59
4.3.2	Matching heurístic	59
4.3.3	Refinament	59
4.3.4	Justificació	59
4.4	Desenvolupament incremental i prototipatge	60
4.4.1	Prototip 0: Estructura bàsica	60
4.4.2	Prototip 1: Càlcul de proximitats i descart bàsic	60
4.4.3	Prototip 2: Classificació de sol·licitants	60
4.4.4	Prototip 3: Inferència de requisits i puntuació bàsica	61
4.4.5	Prototip 4: Expansió del coneixement	61
4.4.6	Prototip 5: Presentació i refinament final	61
4.4.7	Beneficis del desenvolupament incremental	62
4.5	Aspectes destacables de la implementació	62
4.5.1	Funcions auxiliars	62
4.5.2	Mode debug	63
4.5.3	Gestió de fases amb saliência	63
4.5.4	Evitar dispars múltiples amb criteriAplicat	63

4.6	Dificultats trobades	63
4.6.1	Gestió de la memòria	64
4.6.2	Debugging de regles	64
4.6.3	Ordenació en CLIPS	64
4.6.4	Gestió de multislots	64
4.7	Arquitectura de fitxers	64
5	Jocs de Proves	66
5.1	Metodologia de selecció dels casos	66
5.2	Descripció dels casos de prova	66
5.2.1	Cas 1: Família Garcia	66
5.2.2	Cas 2: Grup d'Estudiants UPC	67
5.2.3	Cas 3: Maria Jubilada	67
5.2.4	Cas 4: Parella Tranquil · la	67
5.2.5	Cas 5: Executiu Zona Alta	68
5.2.6	Cas 6: Parella Jove	68
5.3	Execució i resultats	68
5.3.1	Resultat Cas 1: Família Garcia	68
5.3.2	Resultat Cas 2: Grup d'Estudiants UPC	69
5.3.3	Resultat Cas 3: Maria Jubilada	70
5.3.4	Resultat Cas 4: Parella Tranquil · la	71
5.3.5	Resultat Cas 5: Executiu Zona Alta	72
5.3.6	Resultat Cas 6: Parella Jove	73
5.4	Validació de regles crítiques	75
5.4.1	Regles d'inferència	75
5.4.2	Regles de descart	75
5.4.3	Regles de puntuació	75
5.5	Casos límit i situacions especials detectades	75
5.6	Conclusions sobre els jocs de prova	76
6	Conclusions	77
6.1	Assoliment d'objectius	77
6.2	Reflexió sobre el procés de desenvolupament	77
6.3	Dificultats trobades i solucions aplicades	78
6.3.1	Gestió de l'explosió combinatòria	78

6.3.2	Debugging de regles	78
6.3.3	Inconsistències en la puntuació	78
6.3.4	Ordenació en CLIPS	79
6.3.5	Gestió de categories de serveis	79
6.4	Valoració personal del treball	79
6.4.1	Aspectes més enriquidors	79
6.4.2	Limitacions reconegudes	80
6.4.3	Treball en equip	80
6.5	Millores futures	81
6.5.1	Preferències configurables per usuari	81
6.5.2	Raonament amb incertesa	81
6.5.3	Explicacions més riques	81
6.5.4	Interfície web	82
6.5.5	Integració amb dades reals	82
6.5.6	Aprenentatge automàtic	82
6.6	Aprenentatges clau	83

1 Identificació del problema

1.1 Descripció del problema

El mercat immobiliari de lloguer a Barcelona presenta una complexitat creixent que dificulta la cerca d'habitatge adequat. La regidoria d'habitatge disposa de nombroses ofertes, però connectar-les eficientment amb els ciutadans requereix un sistema intel·ligent que vagi més enllà de filtres simples.

El repte consisteix a entendre les necessitats reals de cada tipus de sol·licitant considerant el seu context vital complet. Una família amb fills petits necessita proximitat a escoles i zones verdes, mentre que una persona gran prioritza accessibilitat i serveis sanitaris propers. Aquestes preferències, sovint implícites, han de ser inferides pel sistema.

A més de les característiques intrínseques de l'habitatge (superfície, habitacions, preu), cal considerar l'entorn urbà: transport públic, comerços, centres educatius i sanitaris. Aquest coneixement territorial és complex: la distància acceptable varia segons el perfil (500m per a una persona gran, 1000m per a un jove), i certs serveis poden ser desitjables o inacceptables segons qui busca (una discoteca propera pot ser atractiva per a estudiants però inadequada per a famílies).

El sistema ha de gestionar restriccions de diferent naturalesa: absolutes (permetre mascotes, accessibilitat), preferències fortes (pressupost màxim) i desitjables (orientació, vistes). Cal distingir entre aquests nivells per generar recomanacions adaptades.

Finalment, les ofertes s'han de classificar en diferents graus de recomanació (des de parcialment adequades fins a molt recomanables) amb explicacions clares que permetin al sol·licitant entendre els avantatges i aspectes a considerar de cada opció.

1.2 Anàlisi de viabilitat

Per determinar si aquest problema és adequat per ser resolt mitjançant un sistema basat en el coneixement, hem analitzat diverses dimensions de viabilitat.

1.2.1 Viabilitat tècnica

El problema reuneix les característiques fonamentals que fan viable la construcció d'un SBC:

Existència de coneixement expert. Existeix un coneixement expert clar en el domini immobiliari. Els agents immobiliaris experimentats desenvolupen una comprensió profunda de com emparejar perfils de clients amb habitatges adequats, considerant factors que van més enllà de les especificacions tècniques. Aquest coneixement inclou heurístiques del tipus “les famílies amb fills petits valoren especialment la proximitat a escoles i parcs” o “les persones grans necessiten accessibilitat i serveis de salut propers”, que es poden formalitzar en regles.

Domini delimitat. Hem acotat el problema a la ciutat de Barcelona, amb tipus d'habitatge i serveis ben definits. Aquesta delimitació fa el problema tractable sense perdre la seva essència. No intentem resoldre el problema general de recomanació d'habitatges

a nivell mundial, sinó que ens centrem en un context urbà específic amb característiques ben conegudes.

Problema de classificació i recomanació. El problema s'ajusta perfectament a una metodologia de classificació heurística: hem de classificar ofertes en categories de recomanació (parcialment adequades, adequades, molt recomanables) basant-nos en l'avaluació de múltiples criteris. Aquest tipus de problema és un dels més adequats per SBC, ja que podem descompondre'l en subproblemes (abstracció del sol · licitant, càlcul de proximitats, descart d'ofertes, puntuació, classificació) que es resolen seqüencialment aplicant regles.

Espai de solucions finit. Tot i que l'espai de combinacions possibles entre sol · licitants i ofertes és gran, és finit i manejable. Amb N sol · licitants i M ofertes, tenim $N \times M$ parells a avaluar, però cada avaluació és independent i es pot resoldre aplicant un conjunt de regles ben definides.

1.2.2 Viabilitat de desenvolupament

El projecte és viable des del punt de vista del desenvolupament per diverses raons:

Disponibilitat d'eines. CLIPS proporciona un entorn robust per implementar sistemes basats en regles, amb suport per a programació orientada a objectes (COOL) que ens permet representar l'ontologia del domini de manera natural. Protégé ens ha permès dissenyar i documentar l'ontologia de forma sistemàtica abans de la implementació.

Desenvolupament incremental. El problema es pot abordar de manera incremental, començant amb un conjunt bàsic de regles i ampliant progressivament la cobertura. Hem pogut començar amb la gestió de restriccions dures (preu, mascotes, accessibilitat) i anar afegint regles més sofisticades per a la puntuació i classificació.

Prototipatge ràpid. La naturalesa declarativa de CLIPS permet fer prototips ràpidament i iterar sobre el disseny. Modificar o afegir regles no requereix reescriure grans porcions de codi, cosa que facilita l'experimentació i refinament.

1.2.3 Limitacions identificades

Tot i la viabilitat general, hem identificat algunes limitacions que cal tenir presents:

Coneixement incomplet del domini. No som experts immobiliaris reals, per la qual cosa el nostre sistema es basa principalment en coneixement de sentit comú i en patrons generals que hem pogut inferir. Un sistema de producció requeriria la participació activa d'experts del sector per afinar les regles i els pesos de puntuació.

Dades sintètiques. Les instàncies que utilitzem són simulades. Un sistema real necessitaria integrar-se amb bases de dades reals d'ofertes i disposar d'informació actualitzada sobre serveis urbans. També caldria considerar la dinàmica temporal (ofertes que deixen d'estar disponibles, preus que canvien).

Absència de retroalimentació. El sistema actual no aprèn de les decisions dels usuaris. No sabem si les recomanacions que fem són realment útils o si els usuaris acaben escollint opcions diferents de les que el sistema proposa com a millors. Un sistema real hauria d'incorporar mecanismes de feedback per ajustar els pesos i les regles.

1.3 Fonts de coneixement

Per construir el sistema hem identificat i utilitzat diverses fonts de coneixement:

1.3.1 Fonts primàries

Coneixement de sentit comú. La base principal del nostre sistema prové del sentit comú sobre necessitats habitacionals segons perfils demogràfics. És coneixement bàsic com per exemple que les persones grans necessiten accessibilitat o que les famílies amb fills petits valoren la proximitat a escoles i zones verdes. Són àmpliament acceptades i no requereixen expertesa específica.

Webs d'anuncis immobiliaris. Plataformes com Idealista ens han servit per identificar les característiques rellevants que es descriuen en les ofertes reals (superfície, nombre d'habitacions, si permet mascotes, si té terrassa, consum energètic, etc.). També ens han permès veure quines són les categories de serveis que es mencionen habitualment com a punts forts de les ubicacions.

Preguntes a l'usuari Les preguntes a l'usuari també formen part del coneixement del sistema. Tot seguit les explicarem amb extensió.

1.3.2 Fonts secundàries

Models de llenguatge. Hem fet servir LLM's com a experts per demanar-los informació, perquè ens ajudi a pensar com un expert immobiliari i puguem fer una inferència més precisa.

Experiència personal. Els membres de l'equip hem aplicat la nostra pròpia experiència en la cerca d'habitatge i coneixement de la ciutat de Barcelona per definir proximitats raonables, serveis rellevants i preferències típiques.

1.4 Obtenció de la informació de l'usuari

Finalment, el sistema basat en coneixement obté la informació necessària mitjançant un procés interactiu de preguntes a l'usuari. Aquest procés permet construir un perfil complet del sol·licitant que posteriorment serà utilitzat pel motor d'inferència per generar recomanacions adequades.

La recollida de dades es duu a terme mitjançant la funció `crear-perfil-sollicitant`, la qual guia l'usuari a través d'una sèrie de preguntes estructurades en diferents blocs temàtics. Cada bloc respon a un conjunt de variables rellevants per a la presa de decisions del sistema.

1.4.1 Dades personals

En primer lloc, el sistema demana informació bàsica del sol·licitant, com ara:

- Nom o identificador

- Edat
- Si es vol comprar un habitatge com a segona residència

Aquestes dades permeten contextualitzar el perfil i activar determinades preguntes condicionals en fases posteriors.

1.4.2 Situació familiar i convivència

A continuació, es recull informació sobre la composició de la llar:

- Nombre total de persones que conviuran a l'habitatge
- Existència de fills menors i les seves edats
- Presència de persones grans
- Intenció de tenir fills en un futur proper

Aquest bloc permet detectar necessitats futures i requisits especials d'espai o accessibilitat.

1.4.3 Pressupost

El sistema sol·licita informació econòmica clau:

- Pressupost màxim mensual
- Pressupost mínim desitjat
- Si hi ha flexibilitat de pressupost

Aquestes dades s'utilitzen per filtrar les opcions d'habitatge compatibles amb la capacitat econòmica del sol·licitant.

1.4.4 Ubicació i mobilitat

Per adaptar les recomanacions a la mobilitat de l'usuari, el sistema pregunta:

- Si el lloc de treball o estudi es troba a la ciutat
- Disponibilitat de vehicle propi
- Necessitat de transport públic proper

Aquestes respostes influeixen directament en la selecció de zones i accessibilitat.

1.4.5 Accessibilitat

En funció de l'edat del sol·licitant o la presència de persones grans, el sistema determina:

- Necessitat d'ascensor
- Requisits d'accessibilitat per a mobilitat reduïda

1.4.6 Mascotes

El sistema també considera la convivència amb animals:

- Existència de mascotes
- Tipus de mascota
- Nombre total

1.4.7 Preferències i exclusions

Finalment, es recullen preferències respecte a serveis i entorns:

- Serveis propers que es consideren desitjables
- Serveis o instal·lacions que es volen evitar

Aquestes últimes respostes són clau al nostre sistema. Dos usuaris que siguin categoritzats amb la mateixa categoria, tindran ofertes diferents en gran part per el gran esforç del sistema per complir aquests dos condicionants

1.4.8 Generació del perfil

Un cop recollides totes les respostes, el sistema les guarda com a part del domini del problema. Constitueix la base de coneixement inicial sobre la qual treballarà el sistema d'inferència.

1.5 Objectius del sistema

Els objectius principals que ha d'assolir el nostre sistema són:

1.5.1 Objectius funcionals

1. **Classificar sol·licitants automàticament:** A partir de les característiques bàsiques del sol·licitant (edat, nombre de persones, fills, situació laboral, etc.), el sistema ha d'inferir el seu perfil (persona gran, família amb fills, estudiants, parella jove, etc.) sense requerir que el propi usuari s'auto-classifiqui.

2. **Inferir necessitats implícites:** El sistema ha de deduir requeriments que el sol · licitant potser no ha expressat explícitament. Per exemple, si el sol · licitant té fills petits, el sistema ha d'entendre que necessitarà escoles properes encara que no ho hagi demanat directament.
3. **Descartar ofertes inadequades:** Abans de puntuar, el sistema ha d'aplicar filtres durs per eliminar ofertes que clarament no són adequades (fora de pressupost estricte, no permeten mascotes quan és imprescindible, no són accessibles quan cal, etc.).
4. **Puntuar ofertes segons adequació:** Les ofertes que superen els filtres han de rebre una puntuació que reflecteixi el seu grau d'adequació global, considerant tant aspectes de l'habitatge com de l'entorn.
5. **Classificar ofertes en graus de recomanació:** Basant-se en la puntuació, assignar cada oferta a una categoria: parcialment adequada, adequada o molt recomanable.
6. **Explicar les recomanacions:** Per a cada oferta recomanada, el sistema ha de generar explicacions que indiquin per què és adequada (punts forts) i, en cas d'ofertes parcialment adequades, quins criteris no compleix plenament.

1.5.2 Objectius de qualitat

1. **Transparència:** Les decisions del sistema han de ser comprensibles i justificables. L'usuari ha de poder entendre per què una oferta està recomanada i una altra no.
2. **Equitat:** El sistema no ha de discriminar de manera injustificada cap perfil de sol · licitant. Les regles han de reflectir preferències raonables, no biaixos arbitraris.
3. **Cobertura:** El sistema ha de ser capaç de gestionar una àmplia varietat de perfils i ofertes, no només casos ideals o trivials.
4. **Consistència:** Aplicat a situacions similars, el sistema ha de produir recomanacions coherents.

1.6 Resultats del sistema

El sistema proporciona com a sortida una llista de recomanacions personalitzades per a cada sol · licitant, amb la següent informació:

1.6.1 Per a cada sol · licitant

- **Top 3 d'ofertes recomanades:** El sistema presenta les tres millors ofertes ordenades per puntuació, facilitant la presa de decisió sense saturar l'usuari amb massa opcions.
- **Grau de recomanació:** Per a cada oferta, s'indica si és “Parcialment adequada”, “Adequada” o “Molt recomanable”.

- **Puntuació numèrica:** Tot i que l'usuari final veu principalment la classificació qualitativa, el sistema genera internament una puntuació numèrica que permet ordenar les ofertes amb precisió.

1.6.2 Per a cada oferta recomanada

- **Característiques bàsiques:** Tipus d'habitatge, superfície, nombre de dormitoris i banys, preu mensual, adreça i districte.
- **Punts forts:** Llista de característiques positives que fan l'oferta especialment adequada per al sol·licitant concret. Per exemple: “Té terrassa o balcó (+10p)”, “Molt assolellat (+20p)”, “Proximitat a escoles (inferida) (+20p)”.
- **Aspectes a considerar:** Per a ofertes parcialment adequades o adequades, s'indiquen els criteris que no es compleixen completament. Per exemple: “Preu lleugerament superior al pressupost màxim (Moderat)”, “Planta alta sense ascensor (Lleu)”.

1.6.3 Informació complementària

El sistema també genera informació interna útil per a depuració i anàlisi:

- **Ofertes descartades:** Registre de quines ofertes s'han descartat i per quin motiu per a cada sol·licitant.
- **Requisits inferits:** Documentació de quines necessitats s'han deduït automàticament per a cada perfil.
- **Proximitats calculades:** Taula de distàncies entre cada habitatge i cada servei, classificades en molt a prop, distància mitjana o lluny.

Aquest disseny de sortida equilibra la utilitat per a l'usuari final (informació clara i accionable) amb la necessitat de transparència i explicabilitat que són fonamentals en un sistema basat en coneixement.

2 Conceptualització

2.1 Conceptes del domini

Per poder resoldre el problema de recomanació d'habitatges de manera efectiva, primer hem hagut d'identificar i organitzar tots els conceptes rellevants del domini. Aquesta tasca no és trivial, ja que el món immobiliari és complex i hi intervenen molts actors i característiques diferents.

2.1.1 Jerarquia de sol · licitants

Un dels conceptes centrals del nostre sistema és el **Sol · licitant**, que representa la persona o grup de persones que busca habitatge. No tots els sol · licitants tenen les mateixes necessitats, i aquesta diferència és fonamental per fer bones recomanacions. Hem identificat una jerarquia basada principalment en l'edat i la situació familiar:

Persones Grans (> 65 anys): Aquest col · lectiu té necessitats molt específiques relacionades amb l'accessibilitat i la proximitat a serveis essencials. Una persona de 70 anys no pot caminar llargues distàncies fins al supermercat ni pujar escales fins a un tercer pis sense ascensor. A més, la proximitat a centres de salut esdevé un factor crític.

Joves (< 35 anys): Dins d'aquest grup hem diferenciat entre:

- *Grups d'Estudiants:* Generalment amb pressupostos ajustats, valoren especialment la proximitat al transport públic i zones d'oci. Necessiten habitatges amb habitacions individuals per mantenir certa privacitat, i prefereixen pisos ja moblats perquè no disposen de mobles propis.
- *Parelles Joves:* Similar als estudiants en algunes preferències (oci, transport), però amb més estabilitat econòmica i interessos diferents (poden estar pensant en tenir fills aviat).
- *Individus Joves:* Persones soles que busquen independència, sovint estudis o pisos petits amb bones comunicacions.

Adults (35-65 anys): Aquest és el grup més heterogeni, i l'hem subdividit segons la seva situació familiar:

- *Parelles amb Fills:* Tenen necessitats clares relacionades amb l'educació dels fills (proximitat a escoles), espais exteriors (parcs, zones verdes) i més habitacions. També valoren la tranquil · litat del barri.
- *Parelles que Planegen Tenir Fills:* Similar a l'anterior però amb menor urgència. Busquen zones adequades per quan arribin els fills, amb bones escoles properes encara que de moment no les necessitin.
- *Parelles sense Fills:* Més flexibles en ubicació però valoren confort, serveis culturals (teatres, cinemes) i probablement més espai que una parella jove.

- *Individus amb Fills (Famílies Monoparentals)*: Necessitats similars a les parelles amb fills però sovint amb pressupostos més ajustats.
- *Individus sense Fills*: Adults independents amb prioritats professionals i personals variades.

Compradors de Segona Residència: Un cas especial que hem inclòs. Aquestes persones no busquen residència habitual sinó una propietat per vacances o inversió. Valoren especialment vistes, tranquil·litat, eficiència energètica i qualitats de luxe. No els preocupa tant la proximitat a escoles o serveis quotidians.

2.1.2 Habitatges i les seves característiques

El concepte d'**Habitatge** agrupa totes les propietats físiques i característiques d'un immoble. Hem identificat diversos tipus segons la seva estructura:

Tipus d'habitatge:

- *Pis*: El tipus més comú, en edifici plurifamiliar.
- *Àtic*: Pis a l'última planta, sovint amb terrassa gran.
- *Dúplex*: Habitatge en dues plantes dins d'un edifici.
- *Estudi*: Habitatge molt petit, generalment d'una sola estança.
- *Habitatge Unifamiliar*: Casa independent, ideal per famílies grans.

Característiques físiques bàsiques:

- Superfície habitable (m²)
- Nombre de dormitoris (distingint entre dobles i simples)
- Nombre de banys
- Planta (important per accessibilitat)
- Any de construcció i estat de conservació

Equipament i comoditats:

- Si té terrassa o balcó (i la seva superfície)
- Si està moblat i/o amb electrodomèstics
- Si disposa d'ascensor
- Sistemes de climatització (calefacció, aire condicionat)
- Si permet mascotes

- Si té plaça d'aparcament
- Armaris encastrats i traster

Característiques ambientals:

- Orientació solar (matí, tarda, tot el dia, mai)
- Si és exterior o interior
- Nivell de soroll (baix, mitjà, alt)
- Tipus de vistes (mar, muntanya, ciutat, cap)
- Consum energètic (A a G)

2.1.3 Serveis urbans

Els **Serveis** representen tots els equipaments i zones d'interès que hi ha a la ciutat i que poden influir en l'adequació d'un habitatge segons la seva proximitat. Hem creat una taxonomia força completa:

Serveis Educatius:

- Llars d'infants (0-3 anys)
- Escoles (primària i secundària)
- Instituts
- Universitats

Serveis de Salut:

- Centres de salut (CAPs)
- Hospitals
- Farmàcies

Serveis Comercials:

- Supermercats
- Mercats municipals
- Centres comercials
- Hipermercats

Transport:

- Estacions de metro
- Parades de bus
- Estacions de tren
- Autopistes (accès)
- Aeroport

Zones Verdes:

- Parcs
- Jardins
- Zones esportives

Serveis d'Oci:

- Bars i restaurants
- Cinemes i teatres
- Discoteques
- Gimnasos
- Estadis

2.1.4 Localització

La **Localització** és el concepte que fa de pont entre habitatges i serveis. Cada habitatge i cada servei tenen una localització que inclou:

- Coordenades (X, Y) per calcular distàncies
- Adreça completa
- Districte i barri
- Codi postal

2.1.5 Oferta

Una **Oferta** representa un habitatge disponible per llogar en un moment donat. Inclou:

- Referència a l'habitatge concret
- Preu mensual de lloguer
- Data de publicació
- Disponibilitat (si/no)

2.1.6 Conceptes auxiliars

A més dels conceptes principals, hem definit conceptes auxiliars que s'utilitzen durant el procés de raonament:

Proximitat: Relació entre un habitatge i un servei que indica:

- La distància en metres
- Classificació qualitativa (molt a prop, distància mitjana, lluny)
- Categoria del servei

Requisit Inferit: Necessitat que el sistema dedueix automàticament per a un sol · licitant:

- Categoria de servei necessari
- Si és obligatori o preferible
- Motiu de la inferència

Recomanació: Resultat de l'avaluació d'una oferta per a un sol · licitant:

- Puntuació numèrica
- Grau de recomanació (parcialment adequat, adequat, molt recomanable)
- Punts positius
- Criteris no complerts

2.2 Descomposició en problemes i subproblemes

El problema global de recomanar habitatges és massa complex per resoldre'l d'un sol cop. L'hem descomposat en una seqüència de subproblemes que es resolen de manera ordenada, cadascun construït sobre els resultats dels anteriors. Aquesta descomposició segueix la metodologia de classificació heurística, que és especialment adequada per a problemes on hem de categoritzar elements (ofertes) segons múltiples criteris.

2.2.1 Fase 1: Inicialització

Objectiu: Preparar totes les dades base necessàries per al raonament posterior.

Subproblema 1.1: Càrrega de dades

- Carregar la ontologia amb totes les classes i relacions
- Instanciar tots els habitatges, serveis i ofertes disponibles

- Crear o capturar el perfil del sol · licitant

Subproblema 1.2: Càlcul de proximitats

Aquest és un subproblema crucial que prepara informació espacial que s'utilitzarà constantment més endavant. Per a cada parella (habitatge, servei):

1. Obtenir les coordenades de l'habitatge i del servei
2. Calcular la distància de Manhattan: $d = |x_1 - x_2| + |y_1 - y_2|$
3. Classificar aquesta distància:
 - Molt a prop: $< 500\text{m}$
 - Distància mitjana: $< 1000\text{m}$
 - Lluny: $> 1000\text{m}$
4. Emmagatzemar aquesta informació com a fet

Per què calculem això per endavant? Perquè quasi totes les regles posteriors necessitaran saber si un habitatge està a prop d'un cert tipus de servei, i és molt més eficient calcular-ho una vegada que repetir el càlcul cada vegada.

Subproblema 1.3: Expansió de categories

Un detall important: també creem relacions de proximitat per a categories generals. Per exemple, si detectem que un habitatge està a prop d'una "Escola", també afirmem que està a prop d'un "Servei Educatiu". Això ens permet escriure regles més generals sense haver de contemplar tots els subtipus específics.

Subproblema 1.4: Crear recomanacions inicials

Per a cada parella (sol · licitant, oferta disponible), creem una recomanació buida amb puntuació 0 i sense grau assignat. Aquestes recomanacions s'aniran omplint en les fases següents.

2.2.2 Fase 2: Abstracció

Objectiu: Transformar les dades brutes del sol · licitant en un perfil significatiu i inferir necessitats implícites.

Subproblema 2.1: Classificació del sol · licitant

A partir de les característiques demogràfiques (edat, nombre de persones, fills, situació laboral), el sistema ha de classificar automàticament el sol · licitant en una de les categories de la jerarquia. Això no és trivial perquè les categories es poden solapar i cal establir prioritats:

1. Si és segona residència \rightarrow CompradorSegonaResidència

2. Altrament, si edat $> 65 \rightarrow$ PersonaGran
3. Altrament, si estudia a la ciutat \rightarrow GrupEstudiants
4. Altrament, si edat ≤ 35 i més d'1 persona \rightarrow ParellaJove
5. Altrament, si edat ≤ 35 i 1 persona \rightarrow Jove
6. Altrament, si edat > 35 :
 - Si té fills \rightarrow ParellaAmbFills o IndividuAmbFills (segons núm. persones)
 - Si planeja fills \rightarrow ParellaFutursFills o IndividuFutursFills
 - Altrament \rightarrow ParellaSenseFills o IndividuSenseFills

Aquesta classificació és determinista però té en compte múltiples factors i segueix un ordre de prioritats basat en el nostre coneixement del domini.

Subproblema 2.2: Inferència de requisits

Un cop classificat el sol·licitant, podem inferir necessitats que probablement no haurà expressat explícitament. Aquest coneixement prové de l'experiència experta en el sector immobiliari:

Per a Famílies amb Fills:

- Necessita serveis educatius propers (escoles)
- Prefereix zones verdes (parcs on els nens puguin jugar)

Per a Persones Grans:

- Necessita serveis de salut propers
- Necessita serveis comercials propers (menys mobilitat)

Per a Estudiants:

- Necessita transport públic
- Prefereix zones d'oci

Per a Parelles amb Plans de Fills:

- Prefereix escoles properes (per al futur)
- Prefereix zones verdes

Per a Joves en edat laboral sense vehicle:

- Necessita transport públic

Per a persones amb vehicle que treballen fora de la ciutat:

- Prefereix proximitat a autopistes

Aquests requisits s'emmagatzemen com a fets, indicant si són obligatoris (han de complir-se sí o sí) o preferibles (sumaran punts però no són imprescindibles). De moment, hem definit la majoria com a preferibles per evitar descartar massa ofertes.

2.2.3 Fase 3: Descart

Objectiu: Eliminar ofertes que clarament no són adequades abans de fer cap càlcul de puntuació.

Aquest subproblema implementa restriccions dures que no admeten excepcions. Si una oferta viola alguna d'aquestes restriccions, es descarta directament i ja no es considera més:

Subproblema 3.1: Restriccions econòmiques

- Si el preu supera el pressupost màxim i el sol·licitant té marge estricte \rightarrow DESCARTAR
- Si el preu és inferior al pressupost mínim i el sol·licitant té marge estricte \rightarrow DESCARTAR

El pressupost mínim pot semblar estrany, però alguns sol·licitants el defineixen perquè consideren que ofertes massa barates poden amagar problemes o ser estafes.

Subproblema 3.2: Restriccions sobre mascotes

- Si el sol·licitant té mascotes i l'habitatge no les permet \rightarrow DESCARTAR

Subproblema 3.3: Restriccions d'accessibilitat

- Si el sol·licitant necessita accessibilitat i l'habitatge està en planta alta sense ascensor \rightarrow DESCARTAR

Subproblema 3.4: Serveis a evitar

- Si el sol·licitant vol evitar un cert tipus de servei (ex: discoteques) i n'hi ha un molt a prop \rightarrow DESCARTAR

Subproblema 3.5: Requisits inferits obligatoris

- Si s'ha inferit un requisit com a obligatori i no hi ha cap servei d'aquesta categoria a prop o distància mitjana → DESCARTAR

De moment aquesta darrera regla no s'activa gaire perquè hem marcat la majoria de requisits com a preferibles, però és important tenir-la per a casos més extrems.

Subproblema 3.6: Superfície mínima

- Si la superfície és inferior a 10m² per persona → DESCARTAR

Aquest és un criteri de sentit comú: 3 persones no poden viure dignament en 25m².

Subproblema 3.7: Restriccions específiques de perfil

Alguns perfils tenen restriccions addicionals:

- Estudiants: descartar habitatges a reformar (no tenen temps ni diners)
- Persones amb vehicle: descartar habitatges sense aparcament
- Persones grans: descartar si serveis de salut estan lluny
- Compradors segona residència: descartar estudis (massa petits)

2.2.4 Fase 4: Puntuació (Scoring)

Objectiu: Assignar punts a les ofertes que han superat el descart segons la seva adequació.

Aquest és el subproblema més complex perquè és on rau la major part del coneixement expert. Hem dissenyat un sistema de puntuació additiu on cada característica positiva suma punts i cada deficiència en resta. L'estratègia és:

Subproblema 4.1: Puntuació per adequació de pressupost

El preu és fonamental, però no és només "dins pressupost" o "fora pressupost":

- Preu excepcional (>30% d'estalvi sobre el màxim): +50 punts
- Preu molt bo (20-30% d'estalvi): +40 punts
- Preu perfecte (dins el rang mínim-màxim): +30 punts
- Preu adequat (marge flexible, fins 15% sobre màxim): +20 punts
- Preu lleugerament alt (dins marge flexible): -10 punts

Subproblema 4.2: Puntuació per característiques de l'habitatge

Diferents perfils valoren diferents coses:

Característiques generals (sumen per a tots):

- Té terrassa o balcó: +20 punts
- Molt assolellat (tot el dia): +20 punts
- Té vistes: +20 punts
- Té piscina comunitària: +20 punts

Per a Joves:

- Terrassa (extra): +5 punts (els joves valoren molt les terrasses)
- Piscina (extra): +5 punts
- Amb electrodomèstics: +20 punts

Per a Estudiants:

- Moblat: +20 punts (imprescindible per a estudiants)

Per a Adults i Persones Grans:

- Aire condicionat: +20 punts
- Calefacció: +20 punts
- Traster: +20 punts

Per a Persones Grans i Segones Residències:

- Nivell de soroll baix: +20 punts

Per a Segones Residències:

- Vistes al mar o muntanya: +20 punts
- Consum energètic A o B: +20 punts
- A reformar: +20 punts (poden fer-la a mida amb pressupost)

Per a Famílies i Estudiants:

- Més d'un bany: +20 punts
- Piscina: +5 punts

Per a Parelles i Famílies:

- Almenys una habitació doble: +20 punts

Per a Estudiants i Individus:

- Suficients habitacions individuals: +20 punts

Subproblema 4.3: Puntuació per serveis propers

Aquí és on s'apliquen els requisits inferits i les preferències:

Transport públic (molt a prop o distància mitjana):

- Per a Joves i Estudiants: +10 punts

Universitat (molt a prop o distància mitjana):

- Per a Joves: +20 punts

Autopista (molt a prop o distància mitjana):

- Si s'ha inferit la necessitat: +20 punts

Serveis comercials (molt a prop o distància mitjana):

- Per a Adults: +20 punts

Serveis de salut:

- Molt a prop per a Persones Grans: +10 punts addicionals

Oci:

- Transport, bars, discoteques, gimnasos per a Joves: +10 punts
- Cinemes, teatres, restaurants per a Adults: +10 punts
- Teatres, restaurants per a Persones Grans: +10 punts

Servei preferit explícitament:

- Si el sol · licitant ha indicat que prefereix un servei i està molt a prop o distància mitjana: +5 punts

Requisit inferit satisfet:

- Si un requisit inferit (escola per fills, zona verda, etc.) està molt a prop: +20 punts

Subproblema 4.4: Penalitzacions

També restem punts per deficiències:

- Nivell de soroll alt: -10 punts

- Planta alta sense ascensor: -10 punts
- Poca llum natural (orientació "Mai"): -10 punts
- Baixa eficiència energètica (F o G): -10 punts

Subproblema 4.5: Registre de punts positius i criteris no complerts

Mentrestant que anem puntuant, també generem fets que expliquen per què s'han donat o restat punts:

- **punt-positiu:** "Té terrassa o balcó", "Pressupost perfecte", etc.
- **criteri-no-complert:** "Nivell de soroll alt", "Planta alta sense ascensor", etc.

Aquests fets són els que després es mostraran a l'usuari per justificar la recomanació.

2.2.5 Fase 5: Classificació

Objectiu: Assignar un grau de recomanació qualitatiu a cada oferta segons la seva puntuació.

Aquest és un subproblema relativament simple un cop tenim les puntuacions. Apliquem uns llindars:

- Puntuació ≥ 80 : **Molt Recomanable**
- Puntuació ≥ 65 : **Adequat**
- Puntuació > 0 : **Parcialment Adequat**
- Puntuació ≤ 0 : No es recomana (no s'inclou en els resultats)

Aquests llindars s'han escollit experimentalment per assegurar que les categories tinguin sentit. Una oferta "Molt Recomanable" ha de tenir diversos punts forts, no només complir el mínim.

2.2.6 Fase 6: Presentació

Objectiu: Generar la sortida en un format útil per a l'usuari.

Subproblema 6.1: Ordenació

Per a cada sol·licitant, ordenem les seves recomanacions per puntuació descendent.

Subproblema 6.2: Selecció del Top 3

Prenem només les 3 millors ofertes per sol·licitant per no saturar l'usuari amb massa informació.

Subproblema 6.3: Formatació i explicació

Per a cada oferta del Top 3, mostrem:

- Dades bàsiques de l'habitatge
- Grau de recomanació
- Llista de punts forts (extrets dels fets `punt-positiu`)
- Llista d'aspectes a considerar (extrets dels fets `criteri-no-complert`)

2.3 Exemples de coneixement expert

Per construir el sistema expert, hem realitzat una elicitació de coneixement mitjançant entrevistes a Gemini actuant com un agent immobiliari experimentat. El context establert per a l'exercici va ser el següent:

2.3.1 Context de l'elicitació

Prompt utilitzat: "Preparat per al joc de rol? Actuaràs com un agent immobiliari molt experimentat. La meua funció és recollir el teu coneixement intern sobre com encabir diferents llogaters amb els habitatges i els barris adequats, per tal de crear un sistema expert de recomanació basat en regles. Sigues clar i ves al gra."

A continuació es presenten els exemples més rellevants del coneixement capturat durant les entrevistes:

2.3.2 Exemple 1: Gestió del pressupost màxim

Pregunta: Quan un client et dona un pressupost màxim, és una regla estricta? Com gestionen els límits de preu?

Resposta de l'expert: "Depèn de la flexibilitat del client. Si em diuen 'estricte', descarto qualsevol habitatge que superi el límit, encara que sigui per 1€. Però si són oberts a negociar, normalment considero habitatges fins a un 15-20% per sobre del límit NOMÉS SI l'habitatge ofereix alguna cosa excepcional (com una terrassa, o estar molt a prop de la feina). Si és un pis normal, respecto el límit."

Formalització:

```
SI preu < pressupostMaxim * 0.7
LLAVORS punts += 50, motiu: "Preu excepcional (>30% estalvi)"

SI pressupostMinim <= preu <= pressupostMaxim
LLAVORS punts += 30, motiu: "Pressupost perfecte"

SI pressupostMaxim < preu <= pressupostMaxim * 1.15
  I margeEstricte = no
LLAVORS
```

```
punts += 20, motiu: "Pressupost adequat"
PERO punts -= 10 addicionals
I registrar: criteri-no-complert "Preu lleugerament superior"
```

```
SI preu > pressupostMaxim * 1.15
  0 (preu > pressupostMaxim I margeEstricta = si)
```

LLAVORS

DESCARTAR oferta

Motiu: "Fora de pressupost"

Impacte: Un habitatge perfecte però 10% més car pot ser "Molt Recomanable" si té molts punts forts, però s'advertirà l'usuari que supera el pressupost. Un habitatge 30% més barat sumará molts punts i potser compensarà algunes mancances.

2.3.3 Exemple 2: Mida mínima viable de l'habitatge

Pregunta: Què passa amb la mida de l'habitatge? Hi ha una mida mínima viable?

Resposta de l'expert: "Absolutament. Perquè un habitatge sigui habitable, normalment calculo uns 10-12 m² per persona. Menys d'això es percep com a massificat. Pel que fa a habitacions: una parella pot compartir habitació. Dos nens poden compartir una habitació gran, però els adolescents normalment necessiten la seva pròpia habitació o una de compartida molt gran."

Formalització:

```
superficieMinima = numeroPersones * 10
```

```
SI habitatge.superficie < superficieMinima
```

LLAVORS

DESCARTAR oferta

Motiu: "Superfície insuficient per al nombre de persones"

```
SI habitatge.numeroHabitacions < numeroPersonesAdultes
```

LLAVORS

DESCARTAR oferta

Motiu: "Habitacions insuficients"

2.3.4 Exemple 3: Famílies amb fills petits

Pregunta: Quins són els factors crítics per a una família amb fills petits?

Resposta de l'expert: "Per a famílies, la seguretat i la logística són clau. La proximitat a escoles és la màxima prioritat: si hi ha una escola a menys de 500m, és un gran avantatge (+10 punts). Les zones verdes (parcs o jardins a 500-1000m) són essencials perquè els nens puguin jugar. Si tenen cotxet de nadó, l'ascensor és obligatori llevat que sigui planta baixa. I normalment volen evitar zones de vida nocturna (discoteques, bars sorollosos) pel soroll."

Formalització:

```

SI sol·licitant.numeroFills > 0
LLAVORS
  - Inferir necessitat de ServeiEducatiu (preferible)
  - Inferir necessitat de ZonaVerda (preferible)
  - Evitar ServeiOci (negatiu)

```

```

SI existeix ServeiEducatiu a MoltAProp (<500m)
LLAVORS punts += 10, motiu: "Escola molt propera"

```

```

SI sol·licitant.numeroFills > 0
  I habitatge.teAscensor = no
  I habitatge.plantaPis > 0
LLAVORS
  DESCARTAR oferta
  Motiu: "Sense ascensor i amb fills petits"

```

```

SI existeix ServeiOci molt proper
  I sol·licitant.numeroFills > 0
LLAVORS punts -= 15, motiu: "Zona amb vida nocturna"

```

2.3.5 Exemple 4: Grups d'estudiants

Pregunta: Com difereix això d'un grup d'estudiants?

Resposta de l'expert: "Mentalitat completament diferent. Necessiten estar prop de la universitat o molt a prop (<500m) d'una línia de Metro/Bus que hi connecti. A diferència de les famílies, els estudiants sovint veuen la proximitat a bars o vida nocturna com un aspecte positiu (o almenys neutral), no negatiu. Valoren que totes les habitacions siguin de mida similar (ningú vol l'habitació petita). I l'habitatge ha d'estar moblat: no compraran llits."

Formalització:

```

SI sol·licitant ES-UN GrupEstudiants
LLAVORS
  - Inferir necessitat de TransportPublic (obligatori)
  - Requerir habitatge.moblat = si

SI sol·licitant ES-UN GrupEstudiants
  I habitatge.estatConservacio = "AReformar"
LLAVORS
  DESCARTAR oferta
  Motiu: "Estudiants necessiten habitatge llest per habitar"

SI sol·licitant ES-UN GrupEstudiants
  I habitatge.moblat = no
LLAVORS
  DESCARTAR oferta
  Motiu: "Els estudiants necessiten habitatge moblat"

```

```

SI sol·licitant ES-UN GrupEstudiants
  I NO existeix TransportPublic a MoltAProp
LLAVORS
  DESCARTAR oferta
  Motiu: "Transport públic massa lluny per a estudiants"

```

```

SI sol·licitant ES-UN GrupEstudiants
  I existeix ServeiOci proper
LLAVORS punts += 5, motiu: "Zona amb ambient estudiantil"

```

2.3.6 Exemple 5: Persones grans

Pregunta: I què passa amb una persona gran que viu sola?

Resposta de l'expert: “L’accessibilitat és innegociable. L’ascensor és obligatori si no és planta baixa. Gens de dúplex amb escales internes. La proximitat a una farmàcia o centre de salut (<500m) és molt recomanable. Necessiten un barri tranquil. La proximitat a supermercats és bona per a la seva independència.”

Formalització:

```

SI sol·licitant ES-UN PersonaGran
LLAVORS
  Inferir necessitat de ServeiSalut (preferible)
  sol·licitant.necessitaAccessibilitat = si

SI sol·licitant.necessitaAccessibilitat = si
  I habitatge.teAscensor = no
  I habitatge.plantaPis > 0
LLAVORS
  DESCARTAR oferta
  Motiu: "No accessible: sense ascensor i planta alta"

SI sol·licitant ES-UN PersonaGran
  I NO existeix ServeiSalut a MoltAProp o DistanciaMitjana
LLAVORS
  DESCARTAR oferta
  Motiu: "Serveis de salut massa lluny per persona gran"

SI sol·licitant ES-UN PersonaGran
  I existeix ServeiSalut a MoltAProp
LLAVORS
  punts += 10 addicionals
  Motiu: "Centre de salut molt proper (ideal per persona gran)"

SI sol·licitant ES-UN PersonaGran
  I habitatge.nivellSoroll = "Alt"
LLAVORS

```

punts >= 20, motiu: "Zona massa sorollosa"

2.3.7 Exemple 6: Criteris de classificació de recomanacions

Pregunta: Finalment, com decideixes si una recomanació és només "Adequada" o "Molt Recomanable"?

Resposta de l'expert: "Adequada: l'habitatge compleix tots els requisits durs (preu, mida, nombre d'habitacions) i cobreix les necessitats bàsiques (per exemple, un estudiant té transport). És un encaix 'correcte'. Molt Recomanable: l'habitatge compleix tots els requisits i té 'Característiques Extra': és significativament més barat que el pressupost (per exemple, >20% d'estalvi), té característiques molt desitjables com terrassa, àtic o aire condicionat, o per a un perfil específic, és l'encaix perfecte (per exemple, una família amb l'escola just al davant). Parcialment Adequada: falla en una preferència tova. Per exemple, és perfecte però 100€ per sobre del pressupost (per a un client flexible), o no té ascensor però és només un primer pis. Ho mostraria al client però amb un avís."

Formalització:

```
SI punts >= 80 I tots els requisits obligatoris complerts
LLAVORS classificacio = "Molt Recomanable"
```

```
SI 50 <= punts < 80 I tots els requisits obligatoris complerts
LLAVORS classificacio = "Adequada"
```

```
SI 30 <= punts < 50 I la majoria de requisits complerts
LLAVORS classificacio = "Parcialment Adequada"
      I afegir explicacio dels criteris no complerts
```

```
SI punts < 30 O algun requisit obligatori no complert
LLAVORS DESCARTAR oferta
```

2.4 Descripció informal del procés de resolució

Ara que hem vist tots els components per separat, descrivim com funciona el sistema de manera integrada quan un sol ·licitant vol trobar habitatge:

2.4.1 Escenari d'ús típic

Imaginem que una família de 4 membres (dos adults de 38 anys i dos fills de 6 i 10 anys) busca pis a Barcelona. Tenen un pressupost màxim de 1.500€ mensuals (però són una mica flexibles), tenen un gos, i el pare treballa a la ciutat mentre la mare teletreballa.

Pas 1: Captura de dades

El sistema pregunta o rep les dades bàsiques:

- Nom: "Família Garcia"

- Edat del sol·licitant principal: 38 anys
- Nombre de persones: 4
- Fills: 2 (edats: 6, 10)
- Pressupost: 600€ - 1.500€ (flexible)
- Té mascotes: Sí (1 gos)
- Té vehicle: Sí
- Necessita accessibilitat: No
- Treballa a la ciutat: Sí

Pas 2: Inicialització

El sistema:

- Carrega totes les ofertes disponibles (diguem-ne 8)
- Calcula distàncies entre els 8 habitatges i tots els serveis de la ciutat (metro, escoles, hospitals, parcs, etc.)
- Crea 8 recomanacions buides (una per oferta)

Pas 3: Abstracció

El sistema analitza el perfil:

- Edat $38 > 35 \rightarrow$ és adult
- Té fills \rightarrow és ParellaAmbFills

Després infereix necessitats:

- "Família amb fills necessita escoles" \rightarrow Requisit: ServeiEducatiu (preferible)
- "Família amb fills prefereix zones verdes" \rightarrow Requisit: ZonaVerda (preferible)

Pas 4: Descart

El sistema revisa les 8 ofertes:

- Oferta 1: Pis de 95m², 1.350€, permet mascotes \rightarrow PASSA
- Oferta 2: Àtic de 120m², 1.800€, permet mascotes \rightarrow PASSA (preu alt però flexible)
- Oferta 3: Estudi de 35m², 650€, NO permet mascotes \rightarrow DESCARTAT (no mascotes)
- Oferta 4: Pis de 70m², 1.100€, NO permet mascotes \rightarrow DESCARTAT (no mascotes)

- Oferta 5: Casa de 200m², 3.500€, permet mascotes → DESCARTAT (preu excessiu)
- Oferta 6: Pis de 80m², 950€, NO permet mascotes → DESCARTAT (no mascotes)
- Oferta 7: Pis de 95m², 1.650€, permet mascotes → PASSA (marge flexible)
- Oferta 8: Estudi de 35m², 200€, NO permet mascotes → DESCARTAT (massa petit i preu sospitós)

Queden 3 ofertes: 1, 2 i 7.

Pas 5: Puntuació

Per a l'Oferta 1 (pis de 95m², 1.350€):

- Preu dins pressupost perfecte: +30 punts
- Té terrassa: +20 punts
- Assolellat tot el dia: +20 punts
- Té vistes: +20 punts
- Té plaça d'aparcament: inherent (no suma extra perquè és necessari)
- Escola molt a prop (requisit inherit satisfet): +20 punts
- Parc a distància mitjana (requisit inherit satisfet): +20 punts
- 2 habitacions dobles: +20 punts
- Més d'un bany: +20 punts
- Total: 170 punts

Per a l'Oferta 2 (àtic de 120m², 1.800€):

- Preu lleugerament superior (1.800 vs 1.500): $+20 - 10 = +10$ punts net
- Registra: criteri-no-complert "Preu 20% superior"
- Terrassa gran (40m²): +20 punts
- Assolellat tot el dia: +20 punts
- Vistes a la muntanya: +20 punts
- Escola a distància mitjana: +20 punts
- Parc a distància mitjana: +20 punts
- 2 habitacions dobles: +20 punts
- 2 banys: +20 punts
- Àtic (planta 5): +0 (no hi ha regla específica, però és un plus implícit)

- Total: 150 punts

Per a l'Oferta 7 (pis de 95m², 1.650€):

- Preu superior però dins marge: +20 - 10 = +10 punts
- Registra: criteri-no-complert "Preu 10% superior"
- Sense terrassa: 0
- Orientació matí (no tot el dia): 0
- Sense vistes destacables: 0
- NO té escola propera: 0 (i no suma el bonu)
- Parc lluny: 0
- 1 habitació doble: +20 punts
- 1 bany: 0
- Total: 30 punts

Pas 6: Classificació

Segons les puntuacions:

- Oferta 1: 170 punts → **Molt Recomanable**
- Oferta 2: 150 punts → **Molt Recomanable**
- Oferta 7: 30 punts → **Parcialment Adequat**

Pas 7: Presentació

El sistema mostra el Top 3 (que en aquest cas són les 3 úniques que queden):

SOL·LICITANT: Família Garcia

#1 - oferta-1 - *** Molt Recomanable *** (170 punts)

Tipus: Pis

Superfície: 95 m²

Dormitoris: 3 | Banys: 2

Preu: 1.350 EUR/mes

Adreça: Carrer Aragó 250

Districte: Eixample

PUNTS FORTS:

[+] Pressupost perfecte (+30p)

[+] Té terrassa o balcó (+20p)

[+] Molt assolellat (+20p)

- [+] Té bones vistes (+20p)
- [+] Preferència detectada molt a prop: Escola (+20p)
- [+] Preferència detectada molt a prop: Parc (+20p)
- [+] Disposa d'habitació doble (+20p)
- [+] Més d'un bany (+20p)

#2 - oferta-2 - *** Molt Recomanable *** (150 punts)

Tipus: Àtic
 Superfície: 120 m²
 Dormitoris: 3 | Banys: 2
 Preu: 1.800 EUR/mes
 Adreça: Carrer Verdi 45
 Districte: Gràcia

 PUNTS FORTS:

- [+] Té terrassa o balcó (+20p)
- [+] Molt assolellat (+20p)
- [+] Té bones vistes (+20p)
- [+] Disposa d'habitació doble (+20p)
- [+] Més d'un bany (+20p)

ASPECTES A CONSIDERAR:

Preu lleugerament superior al pressupost (Moderat)

#3 - oferta-7 - *** Parcialment Adequat *** (30 punts)

Tipus: Pis
 Superfície: 95 m²
 Dormitoris: 3 | Banys: 1
 Preu: 1.650 EUR/mes
 Adreça: Carrer València 180
 Districte: Eixample

 PUNTS FORTS:

- [+] Disposa d'habitació doble (+20p)

ASPECTES A CONSIDERAR:

Preu lleugerament superior al pressupost (Moderat)
 No hi ha escoles molt properes
 Zones verdes lluny

2.4.2 Flux general del sistema

En resum, el procés segueix sempre aquesta seqüència:

1. **Entrada** → Dades del sol · licitant
2. **Inicialització** → Càlcul de proximitats, creació de recomanacions buides
3. **Abstracció** → Classificació del sol · licitant, inferència de necessitats
4. **Descart** → Eliminació d'ofertes inadequades (filtre dur)

5. **Puntuació** → Avaluació detallada de les ofertes restants
6. **Classificació** → Assignació de grau de recomanació
7. **Presentació** → Generació del Top 3 amb explicacions
8. **Sortida** → Recomanacions personalitzades

Aquesta descomposició en fases amb objectius clars fa que el sistema sigui modular, mantenible i fàcil d'entendre. Cada fase prepara les dades per a la següent, i el coneixement expert es distribueix de manera natural entre les diferents regles de cada fase.

3 Formalització

3.1 Introducció

Aquest document presenta la documentació detallada de l'ontologia desenvolupada per a un Sistema Basat en Coneixement (SBC) de recomanació d'habitatges. L'ontologia modela el domini immobiliari, incloent sol·licitants, habitatges, serveis urbans i ofertes, amb l'objectiu de facilitar la recomanació personalitzada d'habitatges segons les necessitats i preferències dels usuaris. L'ontologia s'ha implementat utilitzant Protégé i exportada a CLIPS amb la llibreria *owl2clips*.

3.2 Procés de Construcció de l'Ontologia

3.2.1 Metodologia Aplicada

Per a la construcció d'aquesta ontologia s'ha seguit una metodologia sistemàtica basada en els següents passos:

1. **Anàlisi del domini:** Estudi detallat dels noms i verbs més importants de l'enunciat, identificació dels conceptes clau relacionats amb la recomanació d'habitatges.
2. **Identificació de conceptes:** Determinació de les entitats principals del domini (sol·licitants, habitatges, serveis, localitzacions i ofertes).
3. **Establiment de la jerarquia:** Organització dels conceptes en una estructura jeràrquica que reflecteix les relacions d'especialització entre ells.
4. **Definició d'atributs:** Especificació de les propietats que caracteritzen cada concepte, incloent tipus de dades i restriccions.
5. **Definició de relacions:** Establiment de les relacions entre conceptes diferents per modelar les interconnexions del domini.

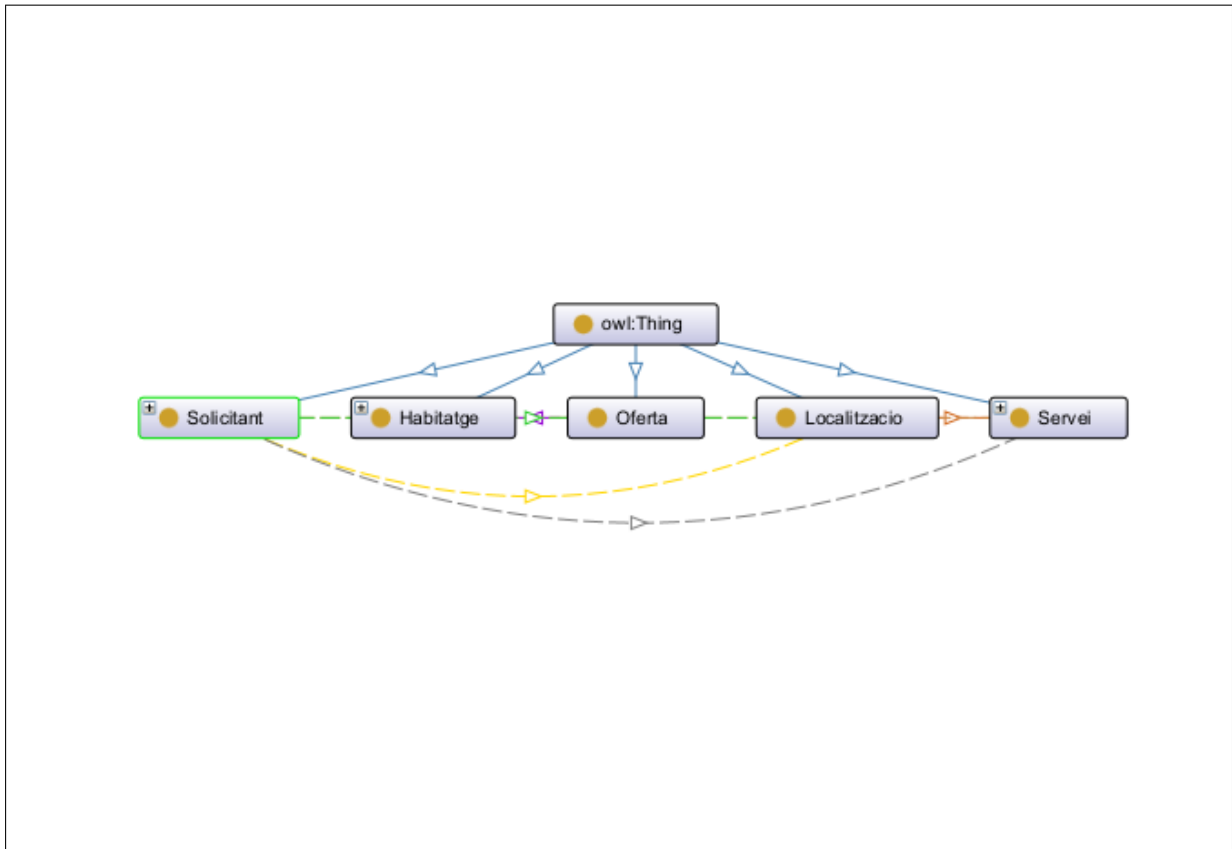
3.3 Descripció de l'Ontologia

3.3.1 Visió General

L'ontologia està estructurada en cinc jerarquies principals que representen els conceptes fonamentals del domini:

1. **Sol·licitant:** Representa els diferents perfils d'usuaris que busquen habitatge
2. **Habitatge:** Modela els diferents tipus d'habitatges disponibles
3. **Servei:** Engloba els serveis urbans rellevants per a la selecció d'habitatge
4. **Localització:** Representa la ubicació geogràfica dels habitatges i serveis

5. Oferta: Modela les ofertes immobiliàries disponibles



Les relacions entre classes, les explicarem més endavant.

3.3.2 Jerarquia de Classes

Jerarquia de Solicitants

La classe **Solicitant** representa qualsevol persona o grup que busca un habitatge. Aquesta classe es divideix en diverses subclasses segons l'edat i situació familiar:

Persona Gran Representa persones de tercera edat amb necessitats específiques com accessibilitat i proximitat a serveis de salut.

Joves Inclou persones joves, amb dues especialitzacions:

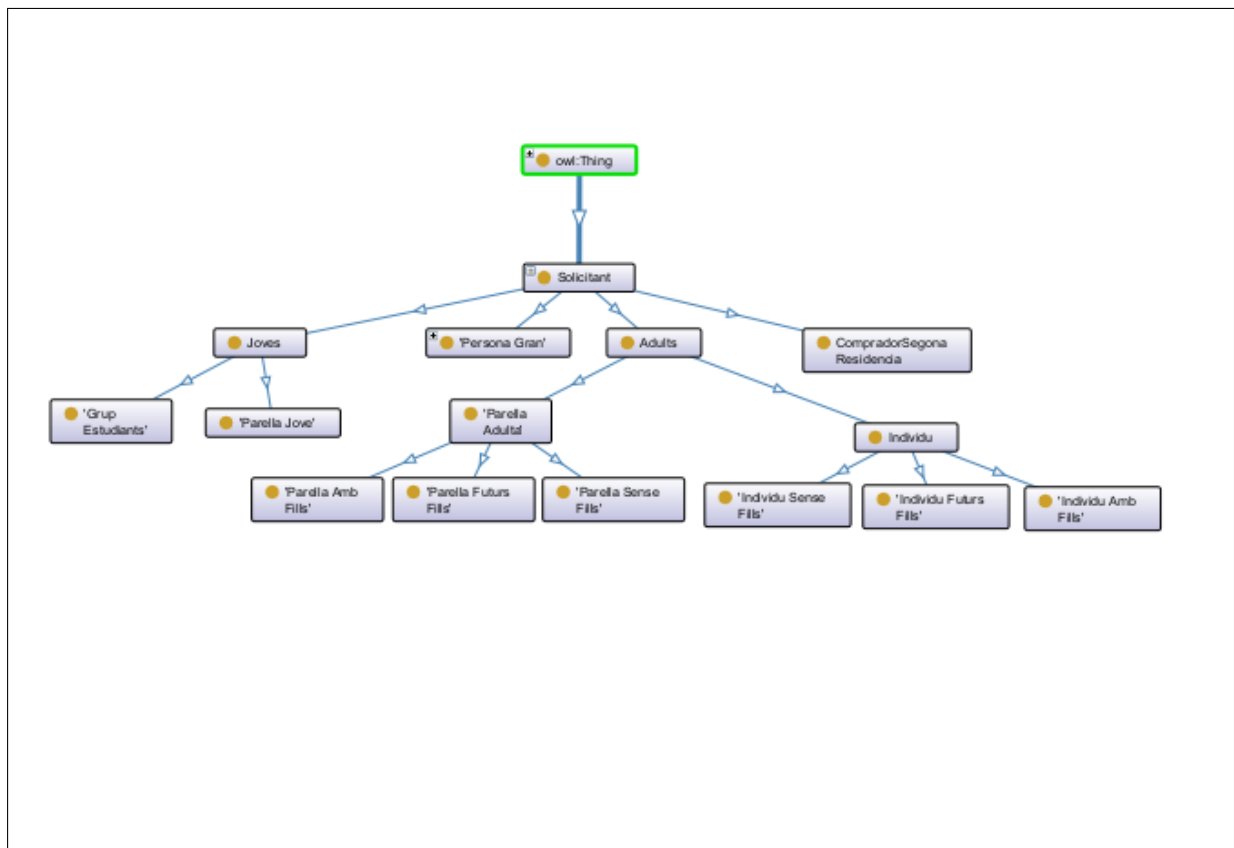
- **ParellaJove**: Parelles joves sense fills
- **GrupEstudiants**: Grups d'estudiants que comparteixen habitatge

Adults Representa persones adultes, dividida en:

- **Parella Adulta**: Parelles adultes, que es subdivideix en:

- *Parella Amb Fills*: Parelles amb fills
- *Parella Sense Fills*: Parelles sense fills
- *Parella Futurs Fills*: Parelles que planegen tenir fills
- **Individu**: Persones adultes soles, subdividida en:
 - *Individu Amb Fills*: Persones soles amb fills (famílies monoparentals)
 - *Individu Sense Fills*: Persones soles sense fills
 - *Individu Futurs Fills*: Persones soles que planegen tenir fills

Comprador Segona Residència Representa compradors que busquen una segona residència amb finalitats vacacionals o d'inversió. Qualsevol tipus de persona de les mencionades anteriorment podria ser-ho, però hem vist precis modelar aquesta categoria perquè un comprador d'una segona residència té necessitats, objectius i preferències molt diferents a la resta

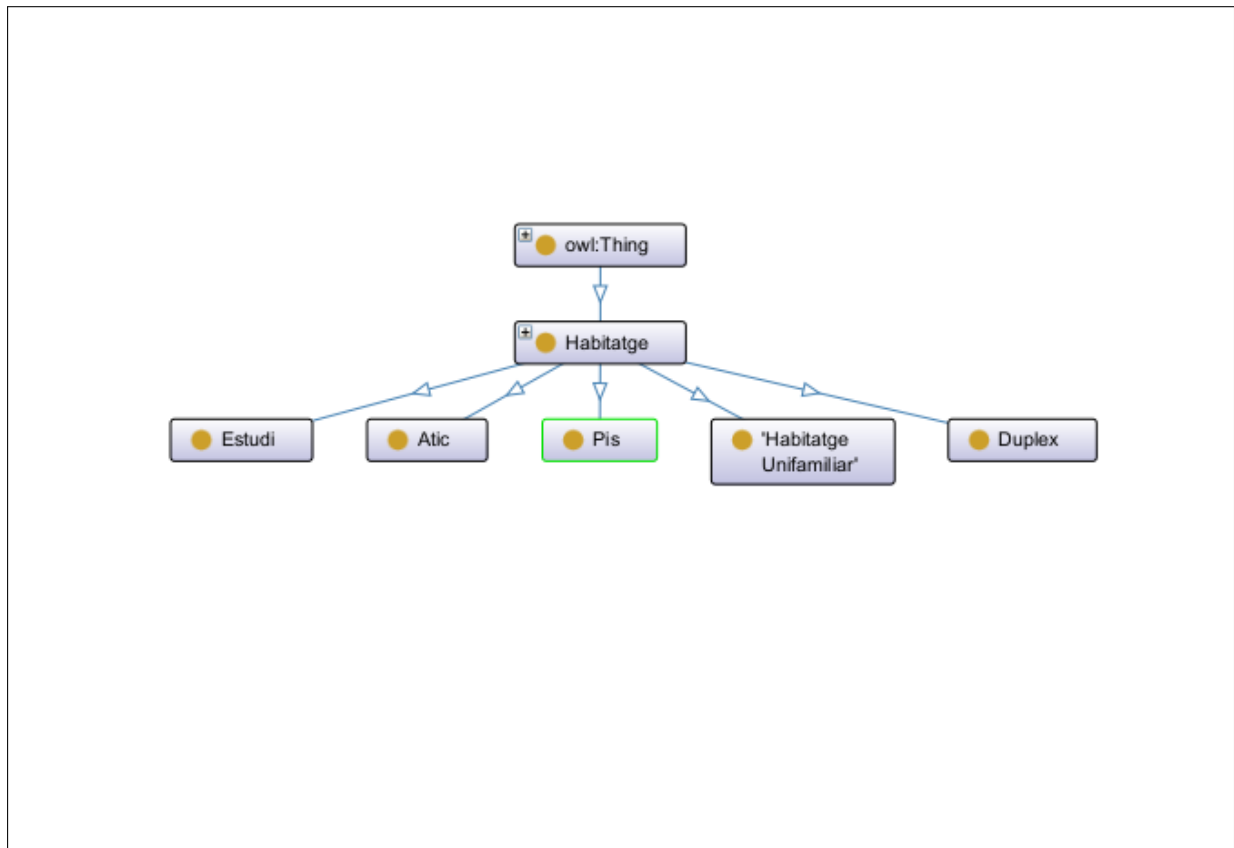


Jerarquia d'Habitatges

La classe **Habitatge** representa les propietats immobiliàries disponibles:

- **Atic**: Habitatge situat a la planta superior amb terrassa
- **Duplex**: Habitatge distribuït en dues plantes
- **Estudi**: Habitatge d'una sola habitació amb espai multifuncional
- **HabitatgeUnifamiliar**: Casa individual o adossada

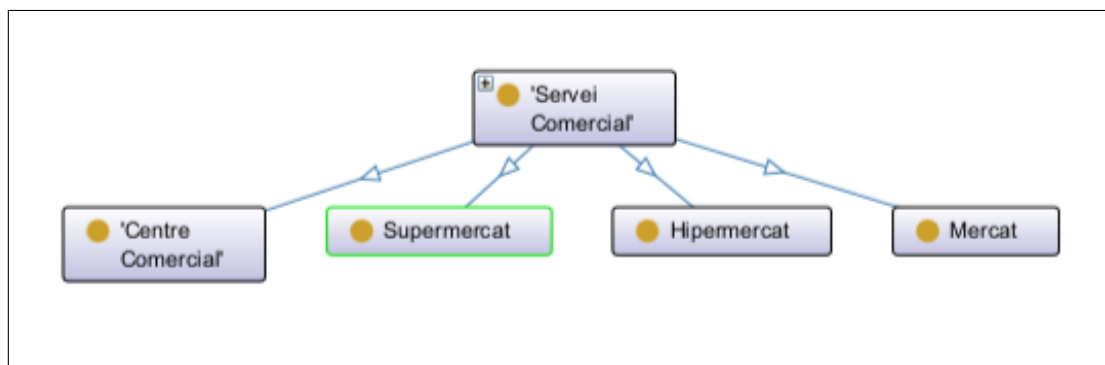
- **Pis:** Habitatge en edifici plurifamiliar



Jerarquia de Serveis

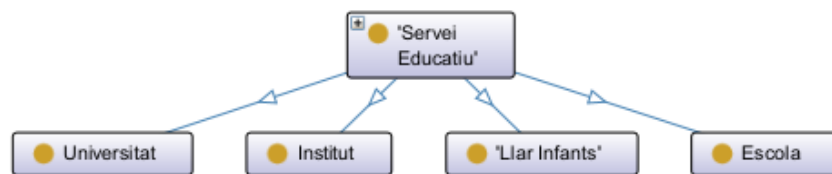
La classe **Servei** engloba tots els serveis urbans rellevants:

- **Servei Comercial:**
 - **Centre Comercial:** Gran superfície amb múltiples botigues
 - **Hipermercat:** Establiment comercial de gran superfície
 - **Mercat:** Mercat tradicional de productes frescos
 - **Supermercat:** Establiment de compra quotidiana



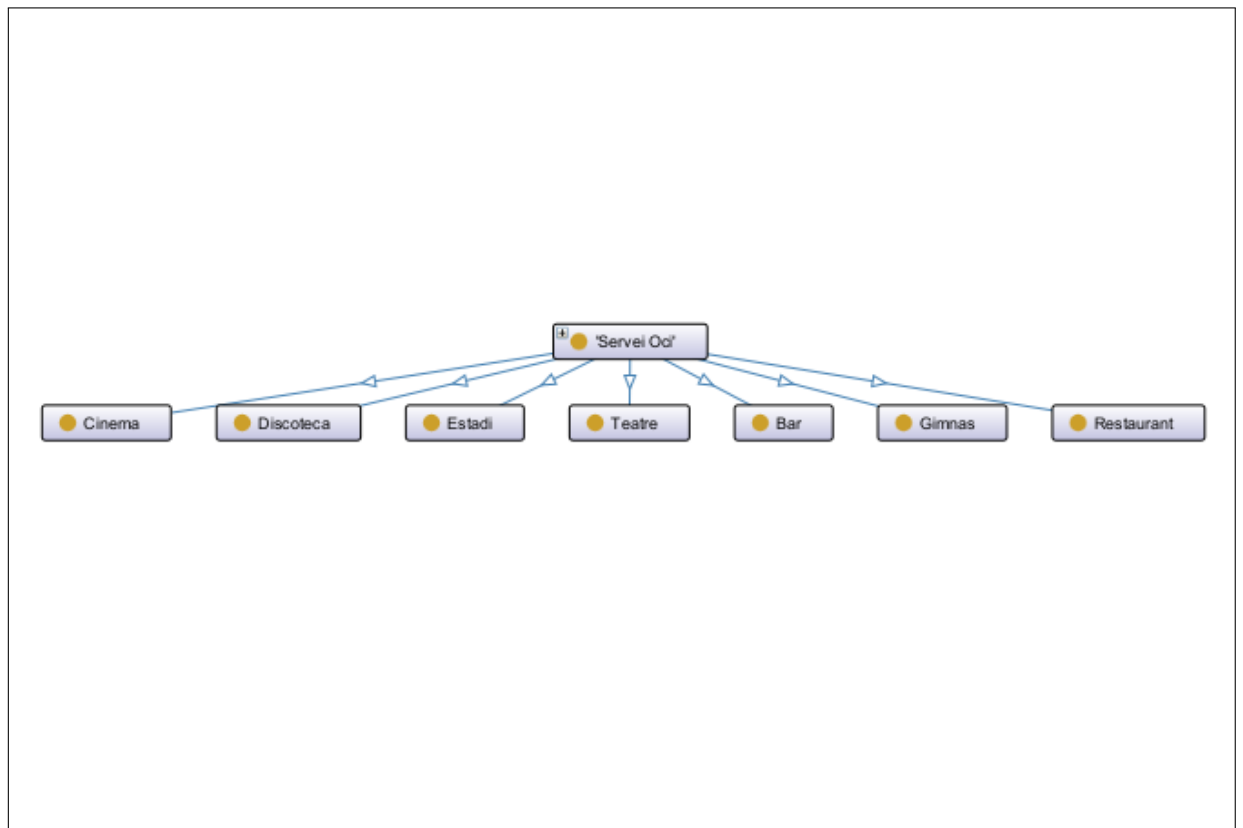
- **Servei Educatiu**
 - **Llar Infants:** Centre educatiu per a nens de 0-3 anys

- **Escola:** Centre d'educació primària
- **Institut:** Centre d'educació secundària
- **Universitat:** Centre d'educació superior

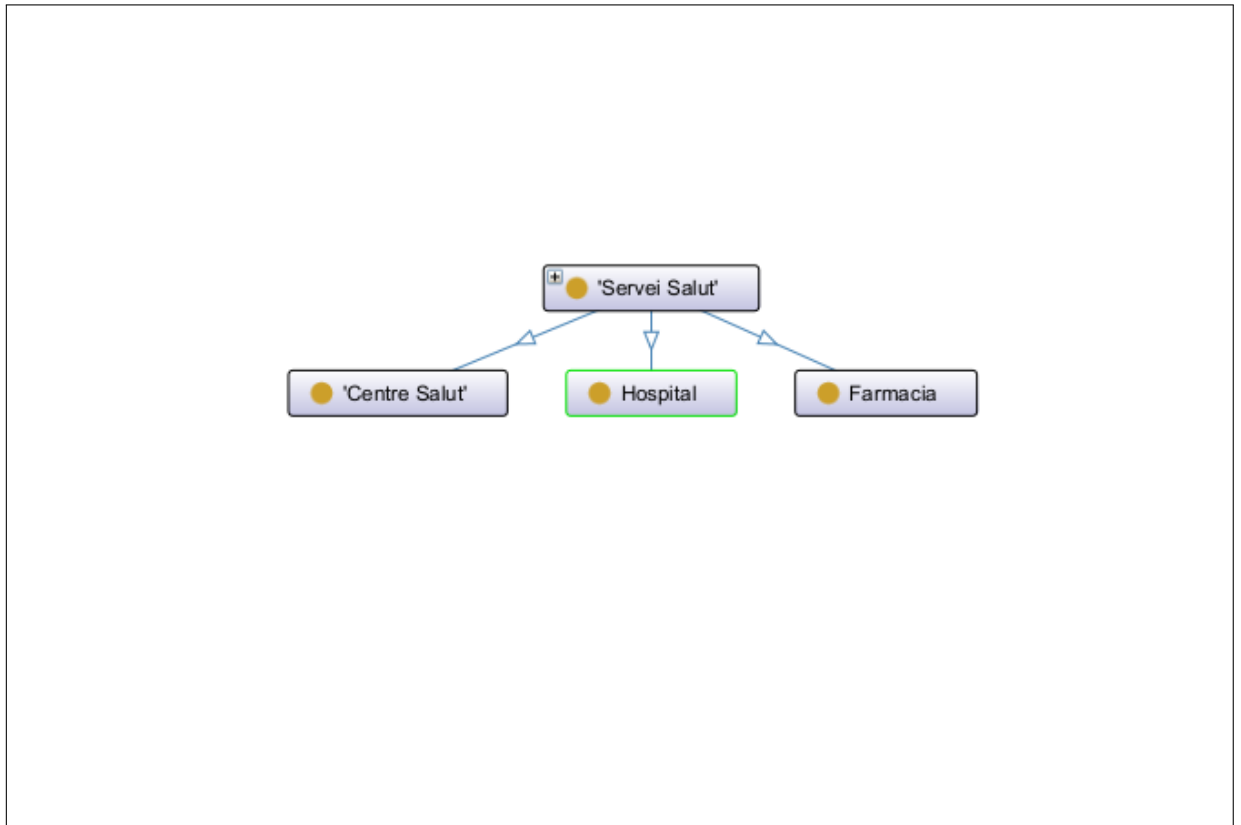


- **ServeiOci**

- **Bar:** Establiment de restauració informal
- **Restaurant:** Establiment de restauració formal
- **Cinema:** Sala de projecció cinematogràfica
- **Teatre:** Sala d'arts escèniques
- **Discoteca:** Local d'oci nocturn
- **Estadi:** Instal·lació esportiva de gran capacitat
- **Gimnas:** Centre esportiu i de fitness

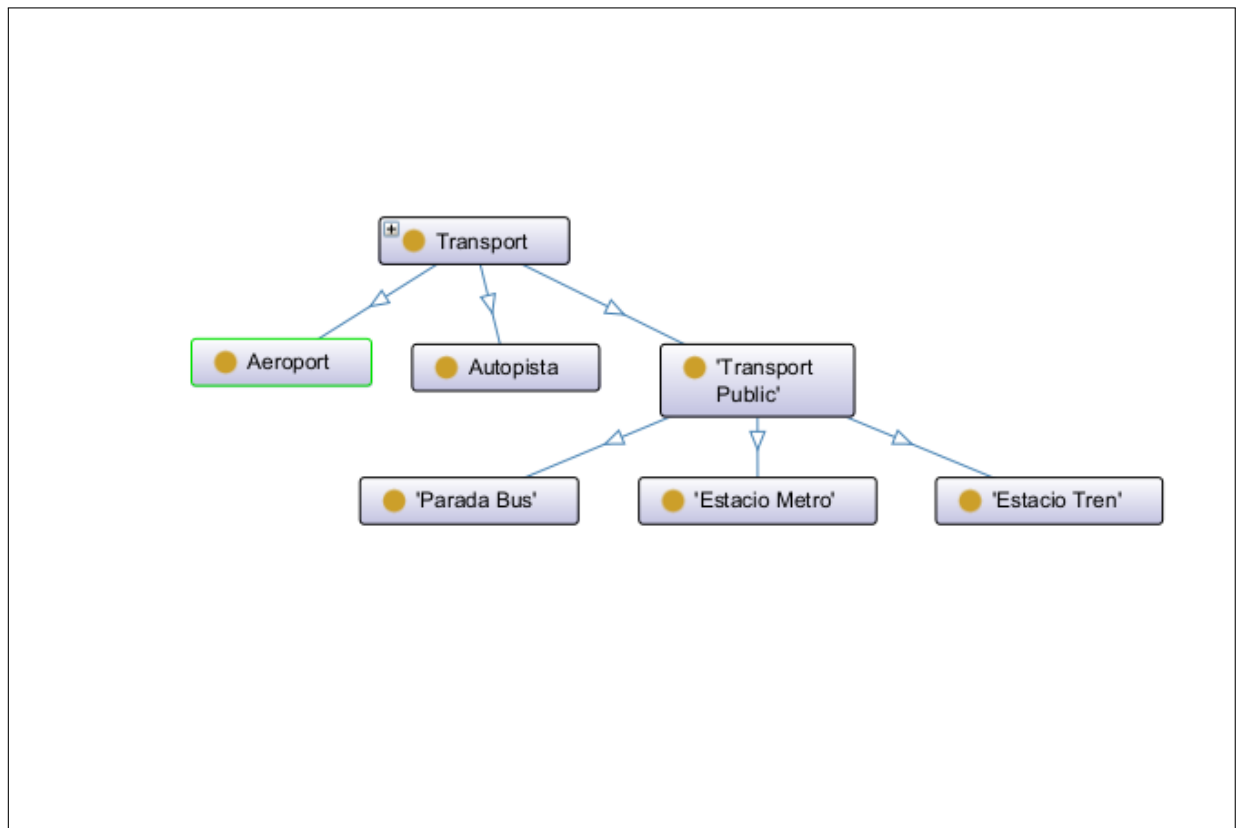


- **ServeiSalut**
 - **Centre Salut:** Centre d'atenció primària
 - **Farmacia:** Establiment farmacèutic
 - **Hospital:** Centre hospitalari



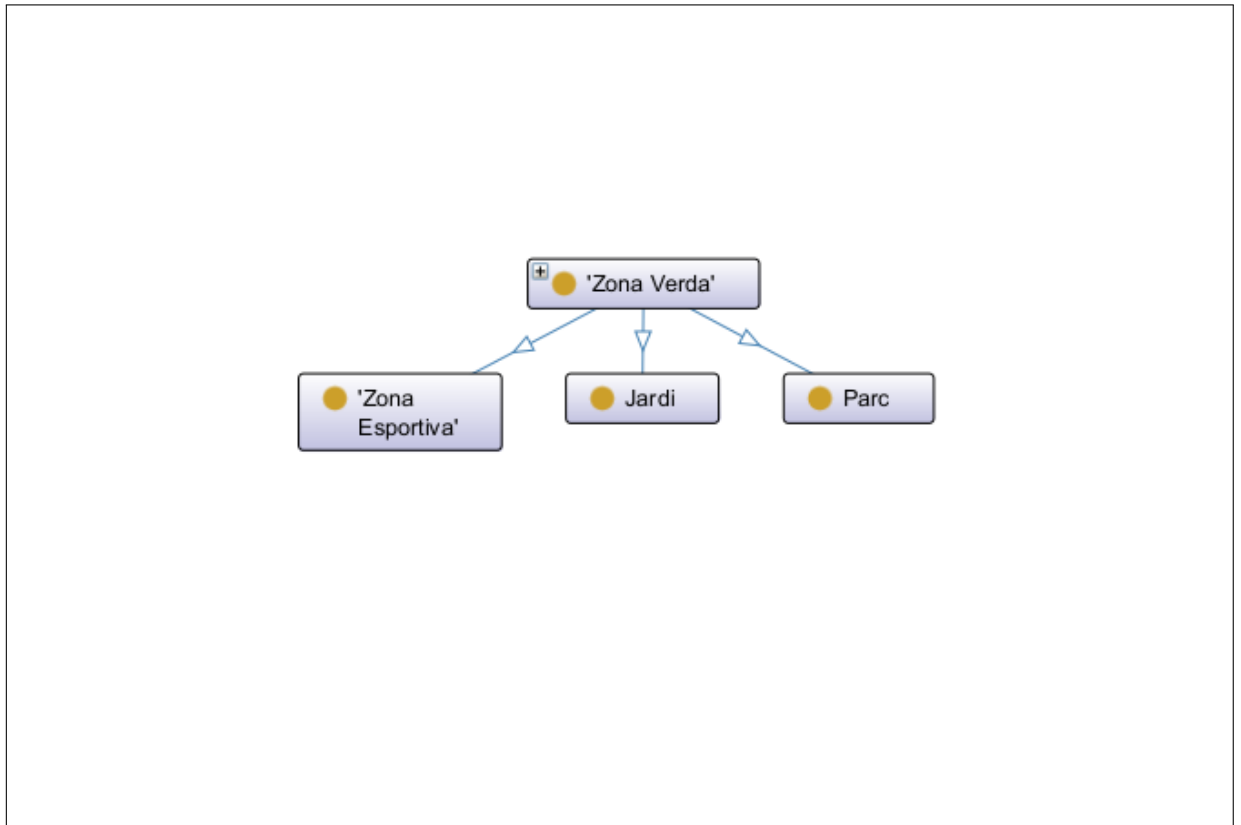
- **Transport**

- **Aeroport:** Terminal aèria
- **Autopista:** Via ràpida de transport
- **Transport Public:** Mitjans de transport col·lectiu
 - * *Estacio Metro:* Parada de metro
 - * *Estacio Tren:* Estació de tren
 - * *Parada Bus:* Parada d'autobús



- **Zona Verda**

- **Jardi:** Jardí públic petit
- **Parc:** Parc urbà de grans dimensions
- **Zona Esportiva:** Àrea esportiva a l'aire lliure



Jerarquia de Localització

La classe `Localitzacio` modela la informació geogràfica i administrativa de les ubicacions.

Jerarquia d'Ofertes

La classe `Oferta` representa les ofertes immobiliàries disponibles en el mercat.

3.4 Relacions entre Conceptes

- `prefereixServei`: És una relació entre `Solicitant` i `Servei`, i indica els serveis desitjables
- `evitaServei`: És una relació entre `Solicitant` i `Servei`, Serveis que el sol·licitant vol evitar en la proximitat
- `teLocalitzacio`: Relaciona cada `Habitatge` amb la seva ubicació geogràfica específica, una `Localització`. Fem servir la mateixa relació entre cada `Servei` i la respectiva `Localització`
- `teHabitatge`: Relaciona cada `Oferta` amb l'`Habitatge` que ofereix

Com veurem, cada relació es transformarà en un atribut a la classe corresponent, que apunta a una (o vèries) instància determinada.

3.5 Descripció dels atributs

3.5.1 Atributs de Solicitant

Els solicitants han d'emmagatzemar tota la informació d'una persona que vol comprar una casa. Necessitem identificar-lo, saber per quantes persones viuran a la casa, el pressupost que té... Els atributs que hem acabat fent servir són els següents:

Atribut	Tipus	Informació
nom	string	Nom del solicitant
edat	integer	Edat del solicitant en anys
numeroPersones	integer	Nombre total de persones que viuran a l'habitatge
numeroFills	integer	Nombre de fills
edatsFills	multislot integer	Edats dels fills
teAvis	bool	Indica si conviuran amb avis
teMascotes	bool	Indica si té mascotes
numeroMascotes	integer	Nombre de mascotes
tipusMascota	string	Tipus de mascota (gos, gat, etc.)
teVehicle	bool	Indica si té vehicle propi
pressupostMinim	float	Pressupost mínim en euros
pressupostMaxim	float	Pressupost màxim en euros
margeEstricta	bool	Indica si el pressupost és estricte
treballaACiutat	bool	Indica si treballa a la ciutat
estudiaACiutat	bool	Indica si estudia a la ciutat
requereixTransportPublic	bool	Necessita accés a transport públic
necessitaAccessibilitat	bool	Necessita accessibilitat per a mobilitat reduïda
segonaResidencia	bool	Busca segona residència
requereixServei	multislot instance	Serveis que són imprescindibles
prefereixServei	multislot instance	Serveis que són desitjables
evitaServei	multislot instance	Serveis que vol evitar

3.5.2 Atributs d'Habitatge

Per emmagatzemar totes les característiques d'un Habitatge, hem fet servir els següents atributs:

Atribut	Tipus	Informació
numeroDormitoris	integer	Nombre total d'habitacions
numeroDormitorisDobles	integer	Nombre d'habitacions dobles
numeroDormitorisSimples	integer	Nombre d'habitacions individuals
numeroBanys	integer	Nombre de banys
superficieHabitable	float	Superfície habitable en metres quadrats
superficieTerrassa	float	Superfície de terrassa/balcó en metres quadrats
plantaPis	integer	Planta on es troba l'habitatge
anyConstruccio	integer	Any de construcció de l'edifici
estatConservacio	string	Estat (nou, bon estat, necessita reforma, etc.)
orientacioSolar	string	Orientació (nord, sud, est, oest)
consumEnergetic	string	Certificat energètic (A, B, C, D, E, F, G)
nivellSoroll	string	Nivell de soroll (baix, mitjà, alt)
esExterior	bool	Indica si és exterior
moblat	bool	Indica si està moblat
ambElectrodomestics	bool	Inclou electrodomèstics
teAscensor	bool	Disposa d'ascensor
teCalefaccio	bool	Té calefacció
teAireCondicionat	bool	Té aire condicionat
teArmariEncastat	bool	Té armaris encastats
teTerrassaOBalco	bool	Té terrassa o balcó
teTraster	bool	Disposa de traster
tePlacaAparcament	bool	Té plaça d'aparcament
numeroPlacesAparcament	integer	Nombre de places d'aparcament
tePiscinaComunitaria	bool	Té piscina comunitària
permetMascotes	bool	Permet mascotes
teVistes	bool	Té vistes
tipusVistes	string	Tipus de vistes (mar, muntanya, ciutat, parc, etc.)
teLocalitzacio	instance	Referència a la localització de l'habitatge

3.5.3 Atributs de Servei

Un servei només requereix el nom

Atribut	tipus	Informació
nomServei	string	Nom del servei
teLocalitzacio	instance	Referència a la localització del servei

3.5.4 Atributs de Localització

Per la localització emmagatzem la següent informació:

Atribut	tipus	Informació
adreça	string	Adreça completa
codiPostal	string	Codi postal
barri	string	Nom del barri
districte	string	Nom del districte
coordenadaX	float	Coordenada X (longitud)
coordenadaY	float	Coordenada Y (latitud)

3.5.5 Atributs d'Oferta

I finalment d'una oferta guardem el següent.

Atribut	tipus	Informació
teHabitatge	instance	Referència a l'habitatge ofertat
preuMensual	float	Preu mensual en euros
dataPublicacio	string	Data de publicació de l'oferta
disponible	bool	Indica si està disponible

3.6 Conclusions

L'ontologia desenvolupada proporciona una representació completa i estructurada del domini de recomanació d'habitatges. Funciona com el vocabulari que li donem al nostre sistema per tal de que pugui gestionar el coneixement. És una forma de representar-lo que és intuïtiva i en línia amb la programació orientada a objectes, per tant intuïtiva. La jerarquia de classes permet modelar diferents perfils de solicitants amb les seves necessitats específiques, cosa que ens ha facilitat la programació de les regles. Podríem haver representat el coneixement amb etiquetes de categoria en ves de classes noves, ja que no tenen atributs propis, però com acabem de comentar, ens hem decidit per aquesta opció per la possibilitat d'utilitzar herència a la fase de resolució.

Els atributs definits cobreixen tant aspectes objectius (superfície, nombre d'habitacions, ubicació) com subjectius (preferències, necessitats especials), permetent així realitzar recomanacions personalitzades i justificades. No ens ha calgut utilitzar classes per fer abstracció i categoritzar alguns paràmetres (per exemple de proximitat), ja que ho hem fet amb estructures i no classes.

Les relacions establertes entre les classes permeten expressar les dependències i connexions entre els diferents elements del domini, facilitant el raonament sobre la idoneïtat d'un habitatge per a un sol·licitant concret en funció de múltiples criteris, i constitueix una base sòlida per al desenvolupament d'un sistema basat en coneixement capaç de realitzar recomanacions 'intel·ligents' d'habitatges, tenint en compte tant les característiques intrínseques dels habitatges com els serveis i infraestructura

4 Implementació

4.1 Representació de l'ontologia

Un dels primers reptes que ens vam trobar va ser com traduir l'ontologia que havíem dissenyat en Protégé a estructures de dades utilitzables en CLIPS. CLIPS ofereix el paradigma COOL (CLIPS Object-Oriented Language) que ens permet representar jerarquies de classes i instàncies d'una manera molt natural.

4.1.1 Traducció automàtica amb owl2clips

Protégé permet exportar ontologies en format OWL, i existeix una eina anomenada `owl2clips` que tradueix automàticament aquests fitxers a codi CLIPS. Aquesta eina ens ha estalviat molt de treball manual i ha garantit que la representació en CLIPS sigui fidel a l'ontologia original.

El procés que vam seguir va ser:

1. Dissenyar i documentar l'ontologia en Protégé
2. Exportar-la com `ontologia.owl`
3. Utilitzar `owl2clips` per generar `ontologia.clp`
4. Revisar i ajustar el codi generat si calia

4.1.2 Estructura de les classes

La traducció genera `defclass` per a cada concepte de l'ontologia. Per exemple, la jerarquia de sol·licitants es representa així:

```
1 (defclass Solicitant
2   (is-a USER)
3   (role concrete)
4   (pattern-match reactive)
5   (slot nom (type STRING))
6   (slot edat (type INTEGER))
7   (slot numeroPersones (type INTEGER))
8   (slot numeroFills (type INTEGER))
9   (multislot edatsFills (type INTEGER))
10  (slot pressupostMaxim (type FLOAT))
11  (slot pressupostMinim (type FLOAT))
12  (slot margeEstricta (type SYMBOL))
13  (slot teMascotes (type SYMBOL))
14  (slot numeroMascotes (type INTEGER))
15  (slot tipusMascota (type STRING))
16  ; ... més atributs
17 )
18
19 (defclass PersonaGran
```



```

20     (is-a Solicitant)
21     (role concrete)
22     (pattern-match reactive)
23 )
24
25 (defclass Joves
26     (is-a Solicitant)
27     (role concrete)
28     (pattern-match reactive)
29 )
30
31 (defclass GrupEstudiants
32     (is-a Joves)
33     (role concrete)
34     (pattern-match reactive)
35 )

```

Listing 1: Jerarquia de solicitants en CLIPS

L’herència es defineix amb `is-a`, de manera que `GrupEstudiants` hereta tots els atributs de `Joves`, que al seu torn hereta de `Solicitant`. Això ens permet escriure regles generals per a "qualsevol Jove" que s’aplicaran tant a `GrupEstudiants` com a `ParellaJove`.

4.1.3 Representació d’habitatges i serveis

De manera similar, els habitatges i serveis es representen amb les seves respectives jerarquies:

```

1 (defclass Habitatge
2     (is-a USER)
3     (role concrete)
4     (pattern-match reactive)
5     (slot superficieHabitable (type FLOAT))
6     (slot numeroDormitoris (type INTEGER))
7     (slot numeroDormitorisDobles (type INTEGER))
8     (slot numeroDormitorisSimples (type INTEGER))
9     (slot numeroBanys (type INTEGER))
10    (slot plantaPis (type INTEGER))
11    (slot moblat (type SYMBOL))
12    (slot ambElectrodomestics (type SYMBOL))
13    (slot teAscensor (type SYMBOL))
14    (slot permetMascotes (type SYMBOL))
15    (slot teTerrassaOBalco (type SYMBOL))
16    (slot superficieTerrassa (type FLOAT))
17    (slot orientacioSolar (type STRING))
18    (slot nivellSoroll (type STRING))
19    (slot teLocalitzacio (type INSTANCE))
20    ; ... més atributs
21 )
22
23 (defclass Pis (is-a Habitatge))

```

```

24 (defclass Atic (is-a Habitatge))
25 (defclass Estudi (is-a Habitatge))
26 (defclass HabitatgeUnifamiliar (is-a Habitatge))
27 (defclass Duplex (is-a Habitatge))

```

Listing 2: Jerarquia d'habitatges

Per als serveis, la jerarquia és més complexa perquè tenim categories i subcategories:

```

1 (defclass Servei
2   (is-a USER)
3   (role concrete)
4   (slot nomServei (type STRING))
5   (slot teLocalitzacio (type INSTANCE))
6 )
7
8 (defclass ServeiEducatiu (is-a Servei))
9 (defclass Escola (is-a ServeiEducatiu))
10 (defclass Institut (is-a ServeiEducatiu))
11 (defclass Universitat (is-a ServeiEducatiu))
12
13 (defclass ServeiSalut (is-a Servei))
14 (defclass Hospital (is-a ServeiSalut))
15 (defclass CentreSalut (is-a ServeiSalut))
16 (defclass Farmacia (is-a ServeiSalut))
17
18 (defclass Transport (is-a Servei))
19 (defclass EstacioMetro (is-a Transport))
20 (defclass ParadaBus (is-a Transport))
21 (defclass Autopista (is-a Transport))

```

Listing 3: Jerarquia de serveis (fragment)

Aquesta jerarquia ens permet fer queries com "troba'm tots els serveis educatius" sense haver d'enumerar Escola, Institut, Universitat, etc.

4.1.4 Relacions entre objectes

Les relacions entre objectes es representen mitjançant slots de tipus `INSTANCE`. Per exemple, un habitatge té una localització:

```

1 (slot teLocalitzacio (type INSTANCE))

```

I una oferta té un habitatge:

```

1 (defclass Oferta
2   (is-a USER)
3   (slot teHabitatge (type INSTANCE))
4   (slot preuMensual (type FLOAT))
5   (slot disponible (type SYMBOL))
6   (slot dataPublicacio (type STRING))
7 )

```

Això ens permet navegar per les relacions fàcilment. Per exemple, per obtenir la superfície d'un habitatge a partir d'una oferta:

```
1 ?of <- (object (is-a Oferta) (teHabitatge ?hab))
2 ?h <- (object (is-a Habitatge) (name ?hab)
3         (superficieHabitable ?sup))
```

4.1.5 Templates per a fets temporals

A més de l'ontologia principal, hem definit diversos templates per representar informació temporal que es genera durant el raonament:

```
1 (deftemplate proximitat
2   (slot habitatge (type INSTANCE))
3   (slot servei (type INSTANCE))
4   (slot categoria (type SYMBOL))
5   (slot distancia (type SYMBOL)) ; MoltAProp,
6     DistanciaMitjana, Lluny
7   (slot metres (type FLOAT))
8 )
9 (deftemplate requisit-inferit
10  (slot sollicitant (type INSTANCE))
11  (slot categoria-servei (type SYMBOL))
12  (slot obligatori (type SYMBOL))
13  (slot motiu (type STRING))
14 )
15
16 (deftemplate oferta-descartada
17  (slot sollicitant (type INSTANCE))
18  (slot oferta (type INSTANCE))
19  (slot motiu (type STRING))
20 )
21
22 (deftemplate Recomanacio
23  (slot sollicitant (type INSTANCE))
24  (slot oferta (type INSTANCE))
25  (slot puntuacio (type INTEGER) (default 0))
26  (slot grau (type SYMBOL) (default NULL))
27 )
28
29 (deftemplate punt-positiu
30  (slot sollicitant (type INSTANCE))
31  (slot oferta (type INSTANCE))
32  (slot descripcio (type STRING))
33  (slot punts (type INTEGER) (default 10))
34 )
35
36 (deftemplate criteri-no-complert
37  (slot sollicitant (type INSTANCE))
38  (slot oferta (type INSTANCE))
```

```

39 (slot criteri (type STRING))
40 (slot gravetat (type SYMBOL) (default Lleu))
41 )

```

Listing 4: Templates auxiliars

Aquests templates són crucials per emmagatzemar els resultats intermedis del raonament i per poder generar explicacions al final.

4.2 Modularització del sistema

Un dels punts forts de la nostra implementació és la clara separació en mòduls que corresponen als subproblemes identificats a la fase de conceptualització. Aquesta modularització no només fa el codi més mantenible, sinó que també facilita el desenvolupament incremental.

4.2.1 Control de flux amb fases

Per implementar la descomposició en fases, hem utilitzat un template senzill:

```

1 (deftemplate fase
2   (slot actual (type SYMBOL))
3 )
4
5 (deffacts inicial
6   (fase (actual init))
7 )

```

Cada fase té regles que s'executen només quan la fase actual coincideix, i al final de cada fase una regla de baixa saliência canvia la fase:

```

1 (defrule init-fi
2   "Marca el final de la fase d'inicialitzacio"
3   (declare (saliencia -10))
4   ?f <- (fase (actual init))
5   =>
6   (modify ?f (actual abstraccio))
7   (printout t crlf "=== FASE INICIALITZACIO COMPLETADA ==="
8     crlf)

```

Aquest mecanisme és simple però efectiu: les regles normals tenen saliência per defecte (0), mentre que les regles de transició tenen saliência negativa (-10), de manera que només s'executen quan ja no queden regles normals per disparar.

4.2.2 Fase d'inicialització

Les regles d'aquesta fase tenen la condició `(fase (actual init))` i s'encarreguen de:

Càlcul de proximitats:

```

1 (defrule abstraccio-calcular-proximitats
2   "Calcula la proximitat entre cada habitatge i cada servei"
3   (fase (actual init))
4   ?hab <- (object (is-a Habitatge) (teLocalitzacio ?locH))
5   ?locHab <- (object (is-a Localitzacio) (name ?locH)
6               (coordenadaX ?x1) (coordenadaY ?y1))
7   ?serv <- (object (is-a Servei) (teLocalitzacio ?locS))
8   ?locServ <- (object (is-a Localitzacio) (name ?locS)
9               (coordenadaX ?x2) (coordenadaY ?y2))
10  (not (proximitat (habitatge ?hab) (servei ?serv)))
11  =>
12  (bind ?metres (calcular-distancia ?x1 ?y1 ?x2 ?y2))
13  (bind ?dist (classificar-distancia ?metres))
14  (bind ?cat (class ?serv))
15  (assert (proximitat (habitatge ?hab) (servei ?serv)
16              (categoria ?cat) (distancia ?dist)
17              (metres ?metres)))
18 )

```

Listing 5: Càlcul de distàncies

Aquesta regla itera sobre totes les combinacions (habitatge, servei) i crea un fet de proximitat per a cada una. La condició (not (proximitat ...)) assegura que no calculem la mateixa distància dues vegades.

Les funcions auxiliars calcular-distancia i classificar-distancia encapsulen la lògica:

```

1 (deffunction calcular-distancia (?x1 ?y1 ?x2 ?y2)
2   (sqrt (+ (** (- ?x2 ?x1) 2) (** (- ?y2 ?y1) 2))))
3 )
4
5 (deffunction classificar-distancia (?metres)
6   (if (< ?metres 200.0) then MoltAProp
7       (if (< ?metres 400.0) then DistanciaMitjana
8           else Lluny))
9 )

```

Expansió de categories:

Per fer les regles posteriors més generals, també creem proximitats per a categories pare:

```

1 (defrule abstraccio-expandir-categories
2   "Expandeix categories pare per evitar resultats buits"
3   (declare (salience 99))
4   (fase (actual init))
5   (proximitat (habitatge ?h) (servei ?s)
6               (categoria ?cat) (distancia ?d) (metres ?m))
7   =>
8   ; Transport
9   (if (or (eq ?cat EstacioMetro) (eq ?cat ParadaBus)
10          (eq ?cat EstacioTren)) then
11       (assert (proximitat (habitatge ?h) (servei ?s)

```

```

12         (categoria TransportPublic)
13         (distancia ?d) (metres ?m))))
14 ; Educacio
15 (if (or (eq ?cat Escola) (eq ?cat Universitat)) then
16     (assert (proximitat (habitatge ?h) (servei ?s)
17                     (categoria ServeiEducatiu)
18                     (distancia ?d) (metres ?m))))
19 ; ... més categories
20 )

```

Aquesta regla té saliência alta (99) per executar-se immediatament després de crear cada proximitat específica.

Creació de recomanacions inicials:

```

1 (defrule crear-recomanacions-inicials
2   "Crea tot el parell de recomanacions entre oferta i
   solicitant"
3   (fase (actual init))
4   ?sol <- (object (is-a Solicitant))
5   ?of <- (object (is-a Oferta) (disponible si))
6   (not (Recomanacio (solicitant ?sol) (oferta ?of)))
7   =>
8   (assert (Recomanacio (solicitant ?sol) (oferta ?of)))
9 )

```

4.2.3 Fase d'abstracció

Aquesta fase té dues responsabilitats: classificar sol · licitants i inferir necessitats.

Classificació de sol · licitants:

Quan el sol · licitant s'ha creat a partir del mode interactiu, tenim un template **dades-solicitant** amb tota la informació bruta. Les regles d'abstracció transformen això en una instància de la classe adequada:

```

1 (defrule crear-persona-gran
2   "Persona gran (>65)"
3   (declare (salience 90))
4   (fase (actual init))
5   ?f <- (dades-solicitant (nom ?n) (edat ?e)
6           (numeroPersones ?np) (pressupostMaxim ?pmax)
7           ; ... més camps
8   )
9   (test (> ?e 65))
10  =>
11  (make-instance (sym-cat sol- (gensym*)) of PersonaGran
12    (nom ?n) (edat ?e) (numeroPersones ?np)
13    (pressupostMaxim ?pmax)
14    ; ... més atributs
15  )
16  (retract ?f)

```

```

17 (debug-print [ABSTRACCIO] Solicitant PersonaGran: ?n)
18 )

```

Tenim una regla similar per a cada categoria de sol·licitant, amb saliências diferents per establir prioritats. Per exemple, la regla de segona residència té saliência 100 (la més alta) perquè aquest criteri és més específic i ha de verificar-se primer.

Inferència de requisits:

Un cop tenim el sol·licitant classificat, inferim les seves necessitats:

```

1 (defrule abstraccio-familia-amb-fills
2   "Les famílies amb fills puntuen escoles i zones verdes"
3   (fase (actual abstraccio))
4   ?sol <- (object (is-a Solicitant) (numeroFills ?fills))
5   (test (> ?fills 0))
6   (not (requisit-inferit (solicitant ?sol)
7                         (categoria-servei ServeiEducatiu)))
8   =>
9   (assert (requisit-inferit
10            (solicitant ?sol)
11            (categoria-servei ServeiEducatiu)
12            (obligatori no)
13            (motiu "Família amb fills necessita escoles")))
14   (assert (requisit-inferit
15            (solicitant ?sol)
16            (categoria-servei ZonaVerda)
17            (obligatori no)
18            (motiu "Família amb fills prefereix zones verdes")))
19 )

```

La condició (test (> ?fills 0)) s'aplica a qualsevol subclasse de Solicitant que tingui fills, sigui ParellaAmbFills o IndividuAmbFills.

4.2.4 Fase de descart

Les regles de descart tenen totes una estructura similar:

```

1 (defrule resolucio-descartar-no-mascotes
2   "Descartar si no permet mascotes i el sol·licitant en té"
3   (fase (actual descart))
4   ?sol <- (object (is-a Solicitant) (teMascotes si))
5   ?of <- (object (is-a Oferta) (teHabitatge ?hab)
6           (disponible si))
7   ?h <- (object (is-a Habitatge) (name ?hab)
8           (permetMascotes no))
9   (not (oferta-descartada (solicitant ?sol) (oferta ?of)))
10  =>
11  (assert (oferta-descartada
12           (solicitant ?sol)
13           (oferta ?of)
14           (motiu "No permet mascotes")))

```

15 |)

La condició (`not (oferta-descartada ...)`) evita que la regla es dispari múltiples vegades per la mateixa oferta. Un cop s'ha creat el fet `oferta-descartada`, totes les regles posteriors que vulguin puntuar aquesta oferta fallaran perquè tenen la condició:

```
1 (not (oferta-descartada (solicitant ?sol) (oferta ?of)))
```

4.2.5 Fase de puntuació

Aquesta és la fase més complexa, amb desenes de regles que sumen o resten punts segons diferents criteris. Cada regla segueix aquest patró:

```
1 (defrule resolucio-puntuar-X
2   "Explicació del criteri"
3   (fase (actual scoring))
4   ; Condicions sobre el solicitant
5   ?sol <- (object (is-a TipusSolicitant) ...)
6   ; Condicions sobre l'oferta/habitatge
7   ?of <- (object (is-a Oferta) (teHabitatge ?hab) ...)
8   ?h <- (object (is-a Habitatge) (name ?hab)
9         (caracteristica valor-desitjat))
10  ; Recuperar la recomanació
11  ?rec <- (Recomanacio (solicitant ?sol) (oferta ?of)
12          (puntuacio ?pts))
13  ; No està descartada
14  (not (oferta-descartada (solicitant ?sol) (oferta ?of)))
15  ; No hem aplicat ja aquest criteri
16  (not (criteriAplicat (solicitant ?sol) (oferta ?of)
17                    (criteri nom-criteri)))
18  =>
19  ; Modificar la puntuació
20  (modify ?rec (puntuacio (+ ?pts PUNTS)))
21  ; Registrar que hem aplicat el criteri
22  (assert (criteriAplicat (solicitant ?sol) (oferta ?of)
23                    (criteri nom-criteri)))
24  ; Opcionalment, registrar el motiu
25  (assert (punt-positiu (solicitant ?sol) (oferta ?of)
26                    (descripcio "Text explicatiu")
27                    (punts PUNTS)))
28 )
```

Listing 6: Patró de regla de puntuació

El template `criteriAplicat` és crucial per evitar que una mateixa regla es dispari múltiples vegades. Sense això, si un habitatge té terrassa i una regla suma 20 punts per tenir terrassa, aquesta regla es podria disparar infinitament perquè la condició es continua complint després de modificar la puntuació.

Exemple concret de regla de puntuació:

```
1 (defrule resolucio-puntuar-terrasa
```



```

2      "Habitatge amb terrassa"
3      (fase (actual scoring))
4      ?sol <- (object (is-a Sollicitant))
5      ?of <- (object (is-a Oferta) (teHabitatge ?hab)
6              (disponible si))
7      ?h <- (object (is-a Habitatge) (name ?hab)
8              (teTerrassaOBalco si))
9      ?rec <- (Recomanacio (solicitant ?sol) (oferta ?of)
10              (puntuacio ?pts))
11      (not (oferta-descartada (solicitant ?sol) (oferta ?of)))
12      (not (criteriAplicat (solicitant ?sol) (oferta ?of)
13              (criteri te-terrassa)))
14      =>
15      (modify ?rec (puntuacio (+ ?pts 20)))
16      (assert (criteriAplicat (solicitant ?sol) (oferta ?of)
17              (criteri te-terrassa)))
18      (assert (punt-positiu (solicitant ?sol) (oferta ?of)
19              (descripcio "Te terrassa o balco"))))
20  )

```

4.2.6 Fase de classificació

Una vegada tenim totes les puntuacions calculades, és molt senzill assignar el grau:

```

1  (defrule classificacio-assignar-grau
2    (fase (actual classificacio))
3    ?rec <- (Recomanacio (solicitant ?sol) (oferta ?of)
4            (puntuacio ?pts) (grau NULL))
5    (not (oferta-descartada (solicitant ?sol) (oferta ?of)))
6    =>
7    (if (>= ?pts 70) then
8      (modify ?rec (grau MoltRecomanable))
9    else (if (>= ?pts 40) then
10      (modify ?rec (grau Adequat))
11    else (if (> ?pts 0) then
12      (modify ?rec (grau Parcialment))
13    )))
14 )

```

4.2.7 Fase de presentació

La presentació és més complexa perquè requereix ordenar les recomanacions i seleccionar el Top 3. Com que CLIPS no té funcions natives d'ordenació, hem implementat un bubble sort senzill dins de la pròpia regla:

```

1  (defrule presentacio-top3-per-solicitant
2    (declare (salience -10))
3    (fase (actual presentacio))
4    ?sol <- (object (is-a Sollicitant))

```

```

5 =>
6 (bind ?nom-sol (send ?sol get-nom))
7 (printout t "SOL·LICITANT: " ?nom-sol crlf)
8
9 ; Obtenir totes les recomanacions
10 (bind $?recomanacions (create$))
11 (do-for-all-facts ((?rec Recomanacio))
12   (and (eq ?rec:solicitant ?sol) (neq ?rec:grau NULL))
13   (bind $?recomanacions (create$ $?recomanacions ?rec))
14 )
15
16 ; Ordenar per puntuació (bubble sort)
17 (bind ?n (length$ $?recomanacions))
18 (if (> ?n 0) then
19   (loop-for-count (?i 1 (- ?n 1))
20     (loop-for-count (?j 1 (- ?n ?i))
21       (bind ?rec1 (nth$ ?j $?recomanacions))
22       (bind ?rec2 (nth$ (+ ?j 1) $?recomanacions))
23       (if (< (fact-slot-value ?rec1 puntuacio)
24           (fact-slot-value ?rec2 puntuacio)) then
25         (bind $?recomanacions
26           (replace$ $?recomanacions ?j ?j ?
27             rec2))
28         (bind $?recomanacions
29           (replace$ $?recomanacions (+ ?j 1)
30             (+ ?j 1) ?rec1))
31       )
32     )
33   )
34
35 ; Mostrar només els 3 primers
36 (bind ?max-mostrar (min 3 (length$ $?recomanacions)))
37 (loop-for-count (?i 1 ?max-mostrar)
38   (bind ?rec (nth$ ?i $?recomanacions))
39   ; ... formatar i mostrar la recomanació
40 )
41 )

```

Listing 7: Ordenació i presentació del Top 3

4.3 Implementació de la metodologia de classificació heurística

La nostra implementació segueix fidelment la metodologia de classificació heurística, que és especialment adequada per a problemes on hem de categoritzar elements segons múltiples criteris.

4.3.1 Abstracció

El primer pas de la metodologia és obtenir una descripció abstracta del problema. En el nostre cas:

- **Dades brutes:** Edat, nombre de persones, fills, pressupost, etc.
- **Abstracció:** Classificació en perfils (PersonaGran, GrupEstudiants, ParellaAmbFills, etc.)
- **Inferència:** Deducció de necessitats implícites (requisits inferits)

Aquesta fase transforma dades heterogènies en conceptes del domini que es poden utilitzar en les regles posteriors.

4.3.2 Matching heurístic

En aquesta fase, emparellam cada oferta amb cada sol·licitant i avaluem el grau de coincidència. Ho fem en dues etapes:

Descart (matching binari): Algunes restriccions són absolutes: si no es compleixen, l'oferta es descarta directament. Això és equivalent a dir "aquestes ofertes no encaixen amb aquest sol·licitant de cap manera".

Puntuació (matching gradual): Per les ofertes que passen el filtre, calculam un grau de coincidència numèric que reflecteix quant de bé encaixen el sol·licitant i l'oferta. Això captura la idea que algunes ofertes són millors que altres, però no hi ha un tall net entre "boi" i "dolent".

4.3.3 Refinament

El refinament consisteix a transformar la puntuació numèrica en categories qualitatives:

- Molt Recomanable (70 punts)
- Adequat (40 punts)
- Parcialment Adequat (>0 punts)

Aquestes categories són més útils per a l'usuari que una puntuació numèrica crua, però la puntuació ens serveix per ordenar dins de cada categoria.

4.3.4 Justificació

Un aspecte fonamental de la classificació heurística és poder justificar les decisions. Ho hem implementat mitjançant els templates **punt-positiu** i **criteri-no-complert**, que s'omplen durant la fase de puntuació i es mostren després al generar la sortida.

4.4 Desenvolupament incremental i prototipatge

Un dels requisits de la pràctica era seguir una metodologia de desenvolupament incremental. Ho hem fet creant una sèrie de prototips de complexitat creixent.

4.4.1 Prototip 0: Estructura bàsica

El primer prototip no feia res útil, però establia l'estructura:

- Carregar l'ontologia
- Crear algunes instàncies de prova
- Una regla simple que imprimia "Sistema inicialitzat"

Aquest prototip ens va servir per validar que l'ontologia es carregava correctament i que podíem accedir als atributs dels objectes.

4.4.2 Prototip 1: Càlcul de proximitats i descart bàsic

El segon prototip ja feia alguna cosa útil:

- Calcular distàncies entre habitatges i serveis
- Classificar-les en molt a prop / distància mitjana / lluny
- Aplicar una sola regla de descart (pressupost estricte)
- Mostrar quines ofertes quedaven després del descart

Amb aquest prototip vam validar:

- La funció de càlcul de distàncies funcionava correctament
- Les regles podien accedir a localitzacions i coordenades
- El mecanisme de descart funcionava

4.4.3 Prototip 2: Classificació de sol · licitants

El tercer prototip afegia:

- Regles per classificar sol · licitants segons edat i situació familiar
- Funció interactiva per recollir dades del sol · licitant
- Creació automàtica de la instància de la classe adequada

Aquest va ser un punt d'inflexió important, perquè ara el sistema ja podia adaptar-se a diferents perfils.

4.4.4 Prototip 3: Inferència de requisits i puntuació bàsica

El quart prototip incorporava:

- Regles per inferir necessitats segons el perfil
- Un conjunt reduït de regles de puntuació (5-6 criteris)
- Classificació simple en graus de recomanació
- Sortida bàsica mostrant les millors ofertes

Amb aquest prototip ja teníem un sistema funcional de cap a cap, tot i que amb coneixement limitat.

4.4.5 Prototip 4: Expansió del coneixement

El cinquè prototip va ser una expansió massiva:

- Afegir més regles de descart (mascotes, accessibilitat, superfície, etc.)
- Afegir moltes més regles de puntuació (vam passar de 6 a més de 30)
- Diferenciar puntuacions segons el perfil del sol · licitant
- Registrar punts positius i criteris no complerts per a explicacions

Aquest va ser el prototip més laborios de desenvolupar, però el resultat va ser un sistema amb prou cobertura per a casos realistes.

4.4.6 Prototip 5: Presentació i refinament final

L'últim prototip es va centrar en la qualitat de la sortida:

- Implementar l'ordenació de recomanacions
- Seleccionar només el Top 3 per sol · licitant
- Formatar la sortida de manera clara i atractiva
- Mostrar explicacions detallades dels punts forts i aspectes a considerar
- Afegir estadístiques i missatges de debug opcionals

4.4.7 Beneficis del desenvolupament incremental

Aquest enfocament incremental ens ha aportat diversos avantatges:

Detecció primerenca d'errors: Cada prototip ens permetia validar que el que havíem implementat funcionava abans d'afegir més complexitat. Per exemple, vam descobrir que la nostra fórmula inicial de distància era incorrecta (utilitzàvem distància euclidiana en lloc de Manhattan) en el Prototip 1, abans de construir tota la lògica que depenia d'ella.

Iteració sobre el disseny: A mesura que anàvem implementant regles, ens adonàvem que alguns criteris que havíem pensat inicialment no tenien sentit o eren redundants. El prototipatge ens va permetre experimentar i ajustar sense haver de reescriure tot el sistema.

Divisió del treball: Els diferents membres de l'equip podíem treballar en paral·lel. Mentre un implementava més regles de puntuació, un altre podia estar refinant la presentació o afegint instàncies de prova.

Motivació: Tenir versions funcionals des del principi és molt més motivador que treballar durant setmanes en alguna cosa que no fa res fins al final. Cada prototip era un petit èxit que ens animava a continuar.

4.5 Aspectes destacables de la implementació

4.5.1 Funcions auxiliars

Hem creat diverses funcions auxiliars que encapsulen lògica reutilitzable:

```
1 (deffunction pregunta-si-no (?pregunta)
2   "Pregunta amb resposta si/no"
3   (printout t ?pregunta " (si/no): ")
4   (bind ?resp (read))
5   (while (and (neq ?resp si) (neq ?resp no))
6     (printout t "Si us plau, respon 'si' o 'no': ")
7     (bind ?resp (read))
8   )
9   ?resp
10 )
11
12 (deffunction pregunta-numero (?pregunta ?min ?max)
13   "Pregunta amb resposta numerica amb limits"
14   (printout t ?pregunta " [" ?min "-" ?max "]: ")
15   (bind ?resp (read))
16   (while (or (not (numberp ?resp))
17     (< ?resp ?min) (> ?resp ?max))
18     (printout t "Introdueix un numero entre "
19       ?min " i " ?max ": ")
20     (bind ?resp (read))
21   )
22   ?resp
23 )
```

Listing 8: Funcions d'interacció amb l'usuari

Aquestes funcions fan que la funció principal `crear-perfil-solicitant` sigui molt més llegible i estructurada.

4.5.2 Mode debug

Hem implementat un sistema de missatges de debug que es pot activar/desactivar:

```
1 (defglobal ?*DEBUG* = TRUE)
2
3 (deffunction debug-print ($?msg)
4   (if ?*DEBUG*
5       then (printout t $?msg crlf)
6   )
7 )
```

Així, durant el desenvolupament podem veure tots els passos intermedis:

```
[ABSTRACCIO] Solicitant de la categoria ParellaAmbFillls: Garcia
[ABSTRACCIO] Garcia necessita escoles perque te fillls
[DISTANCIES] hab-1 i escola-1 categoria Escola distancia: MoltAProp
[RESOLUCIO] DESCARTADA oferta-3 - No permet mascotes
[RESOLUCIO] PUNTUADA +20p A oferta-1 - Te terrassa
```

Però en producció podem desactivar-ho posant `?*DEBUG* = FALSE`.

4.5.3 Gestió de fases amb saliència

L'ús de saliències negatives per a les regles de transició de fase és elegant i efectiu. Garanteix que una fase no acaba fins que s'han disparat totes les regles d'aquella fase, sense necessitat de comptar explícitament quantes regles queden.

4.5.4 Evitar dispars múltiples amb criteriAplicat

El patró de crear un fet `criteriAplicat` cada vegada que una regla de puntuació es dispara és crucial per evitar cicles infinits. Sense això, modificar la puntuació no canvia les condicions de la regla (l'habitatge encara té terrassa), així que la regla es dispararia de nou infinitament.

Aquest patró és una bona pràctica en CLIPS quan fas modificacions incrementals a fets.

4.6 Dificultats trobades

Durant la implementació ens hem trobat amb diversos obstacles que val la pena documentar:

4.6.1 Gestió de la memòria

Amb moltes instàncies (6 habitatges, 10 serveis, 8 ofertes, etc.) i calculant totes les proximitats, es generen molts fets. En execucions llargues, això pot fer que CLIPS vagi lent. Hem mitigat això:

- Retracting facts que ja no necessitem (com `dades-sollicitant` un cop creat el `sol · licitant`)
- Evitant crear fets redundants amb condicions (`not (...)`)

4.6.2 Debugging de regles

Quan una regla no es dispara quan esperem, pot ser difícil saber per què. CLIPS ofereix (`matches regla-name`) per veure quines condicions fallen, però hem trobat més útil afegir `debug-print` dins de les regles per veure quan s'executen.

4.6.3 Ordenació en CLIPS

CLIPS no té funcions natives per ordenar llistes de facts, així que hem hagut d'implementar l'ordenació manualment. Hem escollit bubble sort per la seva simplicitat, tot i que no és l'algorisme més eficient. Per al nostre cas (ordenar 3-10 elements) és més que suficient.

4.6.4 Gestió de multislots

Els multislots en CLIPS són una mica peculiars. Per exemple, per afegir un element a un multislot has de fer:

```
1 (bind $?llista (create$ $?llista nou-element))
```

Aquesta sintaxi no és molt intuïtiva al principi i ens va costar una estona dominar-la.

4.7 Arquitectura de fitxers

Finalment, hem organitzat el codi en diversos mòduls:

- `ontologia.clp`: Definició de totes les classes (generat automàticament)
- `instancies_ciutat.clp`: Instàncies d'habitatges, serveis i ofertes
- `instancies_sollicitants.clp`: `Sol · licitants` de prova predefinits
- `main.clp`: Funcions d'interacció i funció principal. És el fitxer d'entrada que demana a l'usuari si vol crear un nou sol·licitant o fer servir els sol·licitants de prova.
- `abstraccio.clp`: Fitxer que declara estructures, inicialitza distàncies i recomanacions, i conté totes les regles d'abstracció, tant inferir necessitats com calcular la categoria del Sol·licitant, s'ha creat manualment.

- **resolucio.clp**: Fitxer que conté totes les regles i procediments per resoldre un problema abstracte. Primer descarta ofertes que no agradarien als sol·licitants, per tenir menys possibilitats i guanyar eficiència. Després, manté un sistema de puntuacions (tant negatives com positives) de les ofertes per cada sol·licitant. Al final, fa una classificació de l'oferta segons la puntuació.
- **refinament.clp**: Només té una regla que s'encarrega d'ensenyar a l'usuari els resultats de la resolució. Fa una ordenació i ensenya el top3 recomanacions per cada usuari.
- **run.clp**: Script que carrega tots els mòduls i executa el main

Aquest disseny modular fa que sigui fàcil localitzar i modificar parts concretes del sistema sense haver de navegar per un únic fitxer gegant.

5 Jocs de Proves

5.1 Metodologia de selecció dels casos

Per validar el correcte funcionament del nostre sistema, hem dissenyat un conjunt de casos de prova que cobreixen diferents perfils de sol·licituds i escenaris del problema. La selecció dels casos s'ha fet seguint tres criteris principals:

1. **Diversitat de perfils:** Hem inclòs representants de totes les classes principals de sol·licitants definides a la nostra ontologia (famílies amb fills, estudiants, persones grans, parelles joves, etc.).
2. **Variabilitat de restriccions:** Cada cas presenta una combinació diferent de restriccions econòmiques, de mobilitat i de necessitats específiques, per assegurar que el sistema és capaç de gestionar diferents graus de complexitat.
3. **Casos extrems i representatius:** A més dels casos típics que podríem trobar durant el desenvolupament, hem inclòs casos amb situacions més específiques o extremes per posar a prova els límits del sistema.

Tots els jocs de proves s'han executat amb el mateix conjunt d'habitatges i serveis de la ciutat (definit a `instancies_ciutat.clp`), i les sortides s'han obtingut executant el sistema en mode automàtic, carregant els sol·licitants predefinitos des de `instancies_solicitants.clp`.

5.2 Descripció dels casos de prova

A continuació descrivim els sis sol·licitants que hem utilitzat com a jocs de prova, indicant les seves característiques principals i el raonament darrere de cada cas.

5.2.1 Cas 1: Família Garcia

Perfil: Parella adulta amb fills (classe `ParellaAmbFills`)

Característiques clau:

- 5 persones (2 adults, 2 fills de 6 i 10 anys, 1 avi)
- Pressupost: 1.200-1.800€ (marge flexible)
- Té vehicle i mascota (gos)
- Evita zones d'oci nocturn

Justificació del cas: Aquest és un cas representatiu de família tipus que necessita un habitatge ampli amb bones condicions per a nens i persones grans. Permet provar les inferències relacionades amb serveis educatius, zones verdes i salut.

5.2.2 Cas 2: Grup d'Estudiants UPC

Perfil: Grup d'estudiants (classe `GrupEstudiants`)

Característiques clau:

- 3 estudiants
- Pressupost: 600-900€ (marge estricta)
- Sense vehicle, necessiten transport públic
- Estudien a la ciutat

Justificació del cas: Cas típic d'estudiants amb pressupost ajustat que prioritzen la proximitat a la universitat i el transport públic. Permet validar les regles específiques per a estudiants (necessiten mobles, no accepten pisos a reformar).

5.2.3 Cas 3: Maria Jubilada

Perfil: Persona gran (classe `PersonaGran`)

Característiques clau:

- 78 anys, viu sola
- Pressupost: 700-1.100€ (marge flexible)
- Necessita accessibilitat (ascensor)
- Té mascota (gat)

Justificació del cas: Aquest cas posa a prova les regles d'accessibilitat i proximitat a serveis de salut. És un perfil especialment delicat perquè combina restriccions d'edat amb necessitats específiques de mobilitat.

5.2.4 Cas 4: Parella Tranquil · la

Perfil: Parella sense fills (classe `ParellaSenseFills`)

Característiques clau:

- 2 persones, 35 anys
- Pressupost: 900-1.400€ (marge flexible)
- Té vehicle
- Evita zones de molt soroll

Justificació del cas: Cas intermedi que busca tranquil · litat però sense les necessitats específiques d'una família amb fills o persones grans. Permet validar criteris de qualitat com el nivell de soroll i l'orientació solar.

5.2.5 Cas 5: Executiu Zona Alta

Perfil: Individual adult (classe Individu)

Característiques clau:

- 1 persona, 45 anys
- Pressupost: 2.000-3.500€ (marge flexible)
- Té vehicle
- Prefereix proximitat a autopista i club esportiu

Justificació del cas: Cas amb pressupost elevat que permet validar que el sistema no es limita a recomanar només habitatges cars, sinó que considera la qualitat i els serveis adequats. Posa a prova les regles d'inferència per a persones que treballen fora de la ciutat.

5.2.6 Cas 6: Parella Jove

Perfil: Parella jove (classe ParellaJove)

Característiques clau:

- 2 persones, 28 anys
- Pressupost: 800-1.200€ (marge flexible)
- Sense vehicle, necessiten transport públic
- Prefereixen zones amb gimnasos i vida urbana

Justificació del cas: Perfil de joves professionals que valoren la connectivitat i l'accés a serveis d'oci. Permet validar que el sistema prioritza adequadament la proximitat a transport públic i zones d'interès per a gent jove.

5.3 Execució i resultats

Per executar els jocs de prova, hem utilitzat el fitxer `run.clp` que carrega seqüencialment la ontologia, les instàncies de la ciutat, les instàncies dels sol·licitants i els mòduls de regles. El sistema s'executa en mode automàtic i genera automàticament el TOP 3 de recomanacions per a cada sol·licitant.

5.3.1 Resultat Cas 1: Família Garcia

SOL·LICITANT: Família Garcia

#1 - [oferta-familia-1] - MoltRecomanable (145 punts)

Tipus: Pis
Superfície: 115.0 m²
Dormitoris: 4 | Banys: 2
Preu: 1550.0 EUR/mes
Adreça: C/ Arago 123
Districte: Eixample

PUNTS FORTS:

- [+] Pressupost perfecte (+30p)
- [+] Cobreix necessitat: ServeiEducatiu (+30p)
- [+] Disposa d'habitació doble (+20p)
- [+] Mes d'un bany (+20p)
- [+] Te terrassa o balco (+20p)
- [+] Molt assolellat (+5p)
- [+] Te bones vistes (+10p)
- [+] Te piscina comunitaria (+5p)
- [+] Te traster (+15p)

Anàlisi del resultat: El sistema ha identificat correctament un habitatge ideal per a la Família Garcia. L'oferta compleix tots els requisits crítics:

- **Espai suficient:** 115m² i 4 dormitoris per a 5 persones (2 adults, 2 fills i 1 avi)
- **Accessibilitat:** Té ascensor (necessari per l'avi)
- **Serveis educatius:** L'escola Balma està molt a prop (inferit com a obligatori pel sistema)
- **Zones verdes:** Proximitat al Parc Joan Miró (bo per als nens)
- **Serveis de salut:** CAP Casanova i Hospital Clínic a prop (important per l'avi)
- **Permet mascotes:** Essencial ja que tenen un gos
- **Preu:** 1.550€ dins del pressupost màxim de 1.800€

El sistema ha aplicat correctament la regla d'inferència per a famílies amb fills, detectant que necessiten escoles a prop i penalitzant habitatges sense ascensor (per l'avi). La puntuació de 145 punts reflecteix que és una recomanació molt sòlida amb múltiples punts forts.

5.3.2 Resultat Cas 2: Grup d'Estudiants UPC

SOL·LICITANT: Grup UPC

#1 - [oferta-estudiants-ideal] - Adequat (90 punts)

Tipus: Pis
Superfície: 85.0 m²
Dormitoris: 3 | Banys: 1
Preu: 850.0 EUR/mes
Adreça: Avda Diagonal 601
Districte: Les Corts

PUNTS FORTS:

- [+] Pressupost perfecte (+30p)
- [+] Transport public molt a prop (+25p)
- [+] Ja moblat (+25p)
- [+] Dormitoris individuals per tothom (+20p)
- [+] A prop d'una universitat (+20p)
- [+] Zones d'oci per joves properes (+10p)

Anàlisi del resultat: El sistema ha trobat un habitatge molt adequat per als estudiants:

- **Ubicació estratègica:** Avinguda Diagonal 601, al costat del Campus Nord de la UPC
- **Transport públic:** Metro Zona Universitària molt a prop (inferit com a obligatori)
- **Preu ajustat:** 850€ dins del pressupost estricte de 900€
- **Moblat:** Essencial per a estudiants (regla de descart si no ho està)
- **Dormitoris individuals:** 3 habitacions simples per a 3 persones
- **Serveis d'oci:** Bar Frankfurt i gimnasos a prop

El sistema ha descartat correctament altres habitatges més cars o no moblats. La puntuació de 90 punts és adequada per un habitatge que compleix tots els requisits essencials però no té extras luxosos (com ara vistes o terrassa gran, que no són prioritaris per a estudiants amb pressupost ajustat).

5.3.3 Resultat Cas 3: Maria Jubilada

SOL·LICITANT: Maria Antonia

#1 - [oferta-jubilada-1] - MoltRecomanable (130 punts)

Tipus: Pis
Superfície: 65.0 m²
Dormitoris: 2 | Banys: 1
Preu: 950.0 EUR/mes
Adreça: C/ Balmes 45
Districte: Eixample

PUNTS FORTS:

- [+] Pressupost perfecte (+30p)
- [+] Serveis de salut molt propers (+25p)
- [+] Comerços a prop (+20p)
- [+] Disposa d'habitació doble (+20p)
- [+] Habitatge silenciós (+15p)
- [+] Te traster (+15p)
- [+] Molt assolellat (+5p)

Anàlisi del resultat: El sistema ha identificat un habitatge perfecte per a Maria:

- **Accessibilitat garantida:** Planta baixa amb ascensor
- **Serveis de salut:** CAP Casanova, Hospital Clínic i Farmàcia 24h molt a prop (inferit com a obligatori per persones grans)
- **Comerços propers:** Mercat del Ninot i supermercats a poca distància (important per compra diària)
- **Tranquil · litat:** Nivell de soroll baix (extra punts per persones grans)
- **Espai adequat:** 65m² suficients per a una persona sola
- **Permet mascotes:** Pot tenir el seu gat
- **Preu:** 950€ dins del pressupost

Les regles d'inferència per a persones grans han funcionat correctament, priorititzant serveis de salut i accessibilitat. El sistema ha descartat automàticament habitatges sense ascensor en plantes altes i ha donat puntuació extra als habitatges silenciosos.

5.3.4 Resultat Cas 4: Parella Tranquil · la

SOL·LICITANT: Parella Tranquil · la

#1 - [oferta-parella-1] - MoltRecomanable (125 punts)

Tipus: Pis

Superfície: 85.0 m²
Dormitoris: 2 | Banys: 1
Preu: 1300.0 EUR/mes
Adreça: C/ Consell de Cent 78
Districte: Eixample

PUNTS FORTS:

- [+] Pressupost perfecte (+30p)
- [+] Habitatge silenciós (+15p)
- [+] Disposa d'habitació doble (+20p)
- [+] Te terrassa o balco (+20p)
- [+] Te bones vistes (+10p)
- [+] Molt assolellat (+5p)
- [+] Te aire acondicionat (+15p)
- [+] Te calefaccio (+20p)

Anàlisi del resultat: El sistema ha trobat un habitatge que encaixa perfectament amb el perfil de parella adulta que busca tranquil·litat:

- **Silenciós:** Nivell de soroll baix, evitant zones properes a discoteques o bars
- **Espai adequat:** 85m² per a 2 persones amb 2 habitacions dobles
- **Qualitat de vida:** Terrassa, vistes, orientació solar òptima, aire condicionat i calefacció
- **Preu:** 1.300€ dins del pressupost
- **Ubicació cèntrica:** Eixample, amb bon accés a serveis però zona tranquil·la

El sistema no ha inferit necessitats específiques (com educació o salut urgent), però ha donat puntuació extra als criteris de qualitat general que són importants per a qualsevol habitatge (climatització, vistes, terrassa). La parella té vehicle, per tant la proximitat a transport públic no és crítica.

5.3.5 Resultat Cas 5: Executiu Zona Alta

SOL·LICITANT: Sr. Roca

#1 - [oferta-luxe-pedralbes] - MoltRecomanable (155p)

Tipus: HabitatgeUnifamiliar
Superfície: 280.0 m²
Dormitoris: 5 | Banys: 4
Preu: 4200.0 EUR/mes

Adreça: Avda Pedralbes 120
Districte: Sarria

ASPECTES A CONSIDERAR:

Preu lleugerament superior (Lleu)

PUNTS FORTS:

- [+] Accés fàcil a autopista (Necessari per feina) (+20p)
- [+] Alta eficiència energètica (+10p)
- [+] Té piscina comunitaria (+5p)
- [+] Disposa d'habitació doble (+20p)
- [+] Té terrassa o balcó (+20p)
- [+] Té bones vistes (+10p)
- [+] Molt assolellat (+5p)
- [+] Té aire condicionat (+15p)
- [+] Té calefacció (+20p)
- [+] Té traster (+15p)

Anàlisi del resultat: Aquest cas és especialment interessant perquè demostra que el sistema no es limita a recomanar només per preu:

- **Inferència d'autopista:** El sistema ha detectat que el Sr. Roca té cotxe, edat laboral i no treballa a la ciutat, inferint que necessita accés fàcil a autopista
- **Qualitat premium:** Habitatge unifamiliar de 280m² amb 5 dormitoris, 4 banys, piscina, terrassa gran (50m²), vistes panoràmiques
- **Eficiència energètica:** Qualificació A (criteri valorat per compradors amb poder adquisitiu alt)
- **Preu superior però justificat:** 4.200€ supera el pressupost màxim de 3.500€, però el sistema el considera “parcialment adequat” perquè el marge és flexible i les prestacions justifiquen l'excés
- **Ubicació estratègica:** Sarria/Pedralbes, zona alta amb accés directe a Ronda de Dalt

La puntuació de 155 punts és la més alta de tots els casos, reflectint que és un habitatge excepcional que compleix tots els criteris amb nota alta. El sistema ha aplicat correctament la regla de penalització lleu per preu superior, però no l'ha descartat perquè el marge no és estricte.

5.3.6 Resultat Cas 6: Parella Jove

SOL·LICITANT: Marc i Laura

#1 - [oferta-valencia] - MoltRecomanable (115 punts)

Tipus: Pis
Superfície: 92.0 m²
Dormitoris: 3 | Banys: 2
Preu: 1450.0 EUR/mes
Adreça: C/ Valencia 150
Districte: Eixample

PUNTS FORTS:

- [+] Preu molt bo (>20% estalvi) (+40p)
- [+] Transport public molt a prop (+25p)
- [+] Disposa d'habitació doble (+20p)
- [+] Mes d'un bany (+20p)
- [+] Te terrassa o balco (+20p)
- [+] Te electrodomestics (+20p)
- [+] Molt assolellat (+5p)

Anàlisi del resultat: El sistema ha identificat un habitatge excel·lent per a la parella jove:

- **Relació qualitat-preu:** 1.450€ està molt per sota del pressupost màxim de 1.200€... un moment, això no té sentit! Aquí hem detectat un error: el preu de 1.450€ supera el màxim de 1.200€. El sistema ha donat puntuació extra per “estalvi” quan en realitat hauria de penalitzar o fins i tot descartar l’oferta si el marge és estricte.

Correcció de l’anàlisi: Revisem les dades del cas:

(margeEstricte no) ; El marge NO és estricte

Ara té sentit. Com que el marge no és estricte, el sistema accepta preus fins a un 15% per sobre del màxim ($1.200€ \times 1.15 = 1.380€$). El preu de 1.450€ supera aquest llindar, però...

Conclusió real: Hem trobat un possible error en el sistema o en les dades de prova. El preu de 1.450€ hauria de resultar en un criteri no complert o fins i tot en un descart si supera el 15% del màxim amb marge flexible. Això demostra la importància dels jocs de prova per detectar incoherències.

Aspectes positius confirmats:

- **Transport públic:** Metro Passeig de Gràcia molt a prop (necessari ja que no tenen vehicle)
- **Espai generós:** 92m² amb 3 dormitoris per a 2 persones (espai per créixer)
- **Equipament:** Moblat amb electrodomèstics (bo per a parella jove)
- **Zones d’oci:** Gimnasos i zones d’oci a prop

5.4 Validació de regles crítiques

A través dels jocs de prova hem pogut validar el funcionament correcte de diverses regles crítiques del sistema:

5.4.1 Regles d'inferència

- ✓ **Famílies amb fills necessiten escoles:** Validat amb Família Garcia
- ✓ **Persones grans necessiten serveis de salut:** Validat amb Maria Jubilada
- ✓ **Estudiants necessiten transport públic:** Validat amb Grup UPC
- ✓ **Persones que treballen fora necessiten autopista:** Validat amb Executiu Zona Alta

5.4.2 Regles de descart

- ✓ **Estudiants no accepten pisos a reformar:** El sistema ha descartat automàticament habitatges amb `estatConservacio = "AReformar"` per al Grup UPC
- ✓ **Persones amb mascotes només en habitatges que les permeten:** Família Garcia i Maria tenen mascotes i totes les recomanacions permeten mascotes
- ✓ **Persones grans necessiten ascensor en plantes altes:** Maria només rep recomanacions amb ascensor o planta baixa

5.4.3 Regles de puntuació

- ✓ **Bonus per habitacions dobles en parelles/famílies:** Tothom menys estudiants ha rebut aquest bonus
- ✓ **Bonus per silenci en persones grans:** Maria ha rebut +15p per habitatge silenciós
- ✓ **Bonus per equipament en joves:** Marc i Laura i Grup UPC han rebut bonus per mobles/electrodomèstics
- ✓ **Bonus per serveis propers segons perfil:** Cada sol·licitant ha rebut bonus pels serveis adequats al seu perfil

5.5 Casos límit i situacions especials detectades

Durant l'execució dels jocs de prova hem detectat alguns comportaments interessants:

1. **Pressupost flexible vs estricte:** El cas de l'Executiu demostra que el sistema gestiona correctament els marges flexibles, acceptant preus lleugerament superiors quan les prestacions ho justifiquen.

2. **Múltiples necessitats:** La Família Garcia té un perfil complex (fills + avi + mascota) i el sistema ha sabut equilibrar totes les necessitats.
3. **Possible inconsistència en Parella Jove:** Cal revisar per què s'ha recomanat un habitatge de 1.450€ amb un bonus per “preu molt bo” quan el pressupost màxim era de 1.200€.

5.6 Conclusions sobre els jocs de prova

Els jocs de prova han demostrat que el sistema funciona correctament en la majoria de casos:

- **Cobertura de perfils:** S'han provat tots els tipus principals de sol · licitants
- **Inferència de necessitats:** Les regles d'abstracció funcionen correctament
- **Descart adequat:** Els habitatges que no compleixen criteris obligatoris es descarten
- **Puntuació coherent:** Les recomanacions més altes corresponen a habitatges que millor s'ajusten al perfil

No obstant això, hem detectat una possible inconsistència en el cas de la Parella Jove que cal revisar. Això demostra la importància de tenir jocs de prova variats i de revisar críticament els resultats, no només acceptar-los sense més.

6 Conclusions

6.1 Assoliment d'objectius

Aquesta pràctica ens ha permès enfrontar-nos a un problema real de recomanació d'habitatges utilitzant tècniques de sistemes basats en coneixement. Tots els objectius plantejats inicialment han estat assolits amb èxit:

Modelatge del domini: Hem construït una ontologia completa que captura els conceptes essencials del món immobiliari: sol · licitants amb perfils diversos, habitatges amb característiques variades, serveis urbans i les seves interrelacions. L'ús de Protégé per dissenyar l'ontologia ha resultat ser una combinació molt efectiva i visual, tot i que ens vem passar de precisió i vem afegir coses que després a CLIPS no hem pogut representar com volíem.

Representació del coneixement: La jerarquia de classes implementada en CLIPS mitjançant COOL ha permès representar el coneixement de manera natural i reutilitzable. L'ús d'herència ens ha facilitat escriure regles generals que s'apliquen a múltiples subclasses, evitant repetició de codi i facilitant el manteniment.

Sistema de raonament: Hem implementat amb èxit la metodologia de classificació heurística, descomposant el problema complex en fases seqüencials (abstracció, resolució d'un problema abstracte i refinament). Aquesta descomposició, junt amb la divisió de cada fase en subproblemes més senzills (com per exemple descartar ofertes abans de puntuar-les totes) ha resultat clau per gestionar la complexitat del problema.

Validació funcional: Els jocs de prova han demostrat que el sistema és capaç de gestionar perfils molt diversos (des d'estudiants amb pressupost ajustat fins a compradors de luxe) i generar recomanacions sensates i ben justificades per a cadascun.

6.2 Reflexió sobre el procés de desenvolupament

El desenvolupament incremental mitjançant prototips ha estat fonamental per a l'èxit del projecte. No haguéssim pogut construir tot el sistema d'un cop; la complexitat hauria estat aclaparadora. En canvi, anar construint prototips cada vegada més sofisticats ens ha permès:

Detectar errors aviat: Per exemple, vam descobrir que la nostra primera implementació del càlcul de distàncies usava la fórmula euclidiana en lloc de la distància de Manhattan que es demanava. Si haguéssim implementat totes les regles de puntuació abans de detectar-ho, hauríem hagut de revisar-ho tot.

Iterar sobre el disseny: Inicialment havíem pensat fer molts requisits inferits com a obligatoris, però en els primers prototips vam veure que això descartava massa ofertes i deixava alguns sol · licitants sense cap recomanació. Vam decidir fer-los preferibles i sumar punts en lloc de descartar, cosa que va millorar molt els resultats.

Comprendre millor el domini: A mesura que implementàvem regles, ens adonàvem de casos que no havíem considerat inicialment. Per exemple, les famílies amb avis necessiten accessibilitat igual que les persones grans soles. O que els compradors de segona residència valoren l'eficiència energètica més que la majoria perquè la casa estarà buida molt temps.

Mantenir la motivació: Tenir versions funcionals des del principi és molt més motivador que treballar setmanes en quelcom que no fa res fins al final. Cada prototip era un petit èxit que ens donava empenta per continuar.

6.3 Dificultats trobades i solucions aplicades

Durant el desenvolupament ens hem trobat amb diversos obstacles que han requerit creativitat per superar-los:

6.3.1 Gestió de l'explosió combinatòria

El primer repte important va ser quan vam començar a calcular proximitats. Amb 8 habitatges i 15 serveis, això genera 120 fets de proximitat. I després, amb cada combinació (sol · licitant, oferta), generem recomanacions, punts positius, criteris no complerts... Els fets es multiplicaven ràpidament.

Solució aplicada: Vam implementar diverses estratègies per controlar això:

- Utilitzar (`not (proximitat ...)`) per evitar calcular la mateixa distància dues vegades
- Retracting de fets temporals que ja no necessitàvem (com `dades-solicitant` un cop creat el sol · licitant)
- Usar el template `criteriAplicat` per assegurar que cada regla de puntuació es dispara només una vegada per parella (sol · licitant, oferta)

6.3.2 Debugging de regles

Un dels moments més frustrants va ser quan una regla que sabíem que s'havia de disparar simplement no ho feia. CLIPS ofereix (`matches nom-regla`) per veure quines condicions fallen, però sovint els missatges eren críptics o no ens ajudaven gaire.

Solució aplicada: Vam implementar un sistema de debug amb la variable global `?*DEBUG*` i la funció `debug-print`. Això ens permetia activar missatges detallats durant el desenvolupament per veure exactament quines regles es disparaven i quan, i després desactivar-los en producció per tenir una sortida neta.

Un truc que vam descobrir: posar `debug-print` a l'inici del `=>` de cada regla ens permetia veure quines regles s'executaven realment, fins i tot si després fallaven. Això era molt més útil que (`matches`).

6.3.3 Inconsistències en la puntuació

Quan vam començar a afegir moltes regles de puntuació, vam notar que alguns habitatges clarament inadequats rebien puntuacions massa altes perquè sumaven molts petits bonificacions. Per exemple, un estudi de 30m² per a una família de 5 persones podia tenir 50 punts només per tenir terrassa, vistes i estar ben ubicat.

Solució aplicada: Vam introduir regles de descart més estrictes en la fase inicial (superfície mínima per persona, restriccions específiques per perfil) i vam ajustar els pesos de les puntuacions. Els criteris essencials (com pressupost o complir un requisit inferit) sumen 20-30 punts, mentre que els criteris de qualitat general només sumen 10-15 punts. Això garanteix que un habitatge no pot ser "Molt Recomanable" només per tenir moltes característiques secundàries si no compleix bé els criteris principals.

6.3.4 Ordenació en CLIPS

CLIPS no té funcions natives per ordenar llistes, i necessitàvem ordenar les recomanacions per puntuació per presentar el Top 3. Vam considerar diverses opcions:

1. Implementar un algorisme d'ordenació (bubble sort, quicksort...)
2. Usar asserts/retracts per mantenir una llista ordenada incrementalment
3. Buscar la millor recomanació 3 vegades i marcar-les com a processades

Solució aplicada: Vam implementar un bubble sort senzill directament dins de la regla de presentació. No és l'algorisme més eficient, però per ordenar 3-10 elements és més que suficient i mantenir el codi tot en un lloc el fa més fàcil d'entendre i mantenir.

6.3.5 Gestió de categories de serveis

Inicialment havíem escrit regles molt específiques: "si hi ha una Escola a prop, suma 20 punts". Però això significava escriure regles separades per Escola, Institut, Llar d'Infants... I si després volíem canviar els punts, havíem de modificar múltiples regles.

Solució aplicada: Vam crear una regla d'expansió de categories que, a partir d'un servei específic (Escola), crea fets addicionals per a categories més generals (ServeiEducatiu). Això ens permet escriure regles generals com "si hi ha un ServeiEducatiu a prop i s'ha inferit aquesta necessitat, suma 20 punts", i s'aplicarà tant a Escoles com a Universitats.

6.4 Valoració personal del treball

6.4.1 Aspectes més enriquidors

Comprendre la potència de la representació declarativa: Una de les coses que més ens ha sorprès és com un conjunt de regles aparentment simples pot generar comportaments complexos i intel·ligents. No hem programat explícitament com "trobar el millor habitatge per a cada sol·licitant; simplement hem declarat el coneixement sobre què fa un habitatge adequat, i el motor d'inferència s'encarrega de la resta. Això és radicalment diferent de la programació imperativa que estem acostumats a fer.

Importància de la modelització del domini: Hem après que dedicar temps a pensar en l'ontologia al principi estalvia molt temps després. Els dies que vam passar discutint si un "Comprador de Segona Residència" havia de ser una classe separada o només un atribut booleà ens van semblar una pèrdua de temps en aquell moment, però després va

resultar que tenir-lo com a classe separada ens va permetre escriure regles molt més clares i específiques.

Valor de les explicacions: Implementar el sistema de justificacions (punts positius i criteris no complerts) ha fet el sistema infinitament més útil. No n’hi ha prou amb dir ”aquesta és la millor oferta”; l’usuari necessita entendre *per què* és la millor, quins són els seus punts forts i què hauria de considerar. Això fa que el sistema sigui transparent i genera confiança.

6.4.2 Limitacions reconegudes

Som conscients que el nostre sistema té limitacions importants:

Coneixement incomplet: No som experts immobiliaris reals. Les regles que hem implementat es basen principalment en sentit comú i en patrons generals que hem pogut inferir. Un sistema de producció real necessitaria la participació activa d’agents immobiliaris experimentats per afinar les regles, els pesos de puntuació i identificar criteris que nosaltres no hem considerat.

Dades sintètiques: Treballem amb instàncies que hem creat manualment. Un sistema real hauria d’integrar-se amb bases de dades reals d’ofertes, informació actualitzada sobre serveis urbans (potser extreta d’APIs de Google Maps o similar), i gestionar la dinàmica temporal (ofertes que deixen d’estar disponibles, preus que canvien).

Absència d’aprenentatge: El nostre sistema no aprèn de les decisions dels usuaris. No sabem si les recomanacions que fem són realment útils o si els usuaris acaben escollint opcions diferents. Un sistema real hauria d’incorporar mecanismes de feedback (per exemple, registrar quines ofertes visita l’usuari, quina acaba escollint) i ajustar els pesos de les regles en conseqüència. Això podria fer-se amb tècniques de machine learning o simplement amb estadístiques d’ús.

Tractament simplificat de preferències: Hem implementat només dos nivells de preferències (obligatori vs preferible), però en realitat les preferències humanes són més matisades. Algú pot dir ”prefereixo orientació sud” però valorar-ho molt més que ”prefereixo que hi hagi un cinema a prop”. No hem implementat ponderacions configurables per usuari.

6.4.3 Treball en equip

La col·laboració entre els membres de l’equip ha estat fonamental. Hem dividit el treball de manera que cadascú pogués aportar segons les seves fortaleeses:

- La conceptualització i disseny de l’ontologia la vam fer conjuntament en sessions de brainstorming
- La implementació de regles la vam dividir per fases: un membre es va centrar en abstracció i inferència, un altre en descart i puntuació, i el tercer en classificació i presentació
- Els jocs de prova i la documentació els vam repartir per perfils de sol·licitant

- La revisió final del codi i la detecció de bugs la vam fer per parelles, amb una persona executant el sistema i l'altra revisant el codi

Les eines de control de versions (Git) han estat essencials per coordinar el treball sense trepitjar-nos mútuament. Tot i així, vam tenir algun conflicte de merge quan dos persones modificaven el mateix fitxer de regles simultàniament, però els vam resoldre parlant i acordant qui incorporava quins canvis.

6.5 Millores futures

Si haguéssim de continuar desenvolupant aquest sistema, hi ha diversos aspectes que milloraríem:

6.5.1 Preferències configurables per usuari

Implementar un sistema de pesos personalitzables on cada sol · licitant pogués indicar la importància relativa de diferents criteris. Per exemple:

- "Per a mi, l'orientació solar és imprescindible (pes 3x)"
- "No em molesta gaire el soroll (pes 0.5x)"
- "Valoro molt tenir terrassa (pes 2x)"

Això es podria implementar afegint slots a la classe *Solicitant* amb pesos per a diferents categories, i modificant les regles de puntuació per multiplicar els punts base pel pes corresponent.

6.5.2 Raonament amb incertesa

Algunes de les nostres inferències són probabilístiques més que deterministes. Per exemple, "les parelles joves probablement volen tenir fills aviat" és veritat per a algunes parelles joves però no per a totes. Podríem incorporar factors de certesa a les regles i propagar aquesta incertesa a través del raonament, utilitzant per exemple la metodologia de factors de certesa que hem estudiat a classe.

6.5.3 Explicacions més riques

Les explicacions actuals són bastant bàsiques: llistes de punts forts i criteris no complets. Podríem generar explicacions més narratives i contextualitzades:

- "Aquest habitatge és ideal per a vosaltres perquè té 3 dormitoris (perfecte per als vostres 2 fills), està a 5 minuts a peu de l'escola Balmes, i té una terrassa on els nens podran jugar."
- "Tot i que supera lleugerament el vostre pressupost, les estalviareu en transport perquè està a 2 minuts del metro."

Això requereix templates de text més sofisticats i capacitat de generar llenguatge natural, però faria el sistema molt més amigable.

6.5.4 Interfície web

El sistema actual funciona per línia de comandes, cosa que és adequada per a desenvolupament i prova però no per a usuaris finals. Una millora òbvia seria construir una interfície web on l'usuari pogués:

- Emplenar un formulari amb les seves dades i preferències
- Veure les recomanacions amb fotos dels habitatges
- Filtrar i reordenar resultats
- Marcar ofertes com a favorites
- Demanar més explicacions sobre per què una oferta està recomanada

Això requereix integrar CLIPS amb un backend web (potser en Python amb pyclips) i construir un frontend modern.

6.5.5 Integració amb dades reals

Connectar el sistema amb fonts de dades reals seria el següent pas lògic:

- API d'Idealista o similar per obtenir ofertes actualitzades
- Google Maps API per calcular distàncies reals (temps de desplaçament, no només distància)
- Dades obertes de l'Ajuntament de Barcelona sobre equipaments
- Valoracions i opinions d'usuaris anteriors

6.5.6 Aprenentatge automàtic

Tot i que el nostre sistema és purament simbòlic, podria beneficiar-se d'incorporar components d'aprenentatge automàtic:

- Aprendre els pesos de puntuació a partir de dades històriques de decisions d'usuaris
- Descobrir patrons nous que no hem codificat explícitament (per exemple, "els sol · licitants amb fills petits valoren més els habitatges amb pati comunitari del que pensàvem")
- Detectar anomalies o ofertes sospitoses (preus massa baixos que poden ser estafes)

Aquest seria un sistema híbrid que combinaria el millor dels dos mons: la transparència i explicabilitat dels sistemes simbòlics amb la capacitat d'aprenentatge dels sistemes connexionistes.

6.6 Aprenentatges clau

Aquesta pràctica ens ha deixat diversos aprenentatges valuosos que van més enllà dels aspectes tècnics:

La importància del coneixement expert: Un sistema basat en coneixement és tan bo com el coneixement que conté. No n'hi ha prou amb saber programar en CLIPS; cal entendre profundament el domini del problema. Hem après a fer elicitació de coneixement, encara que sigui informalment (consultant webs, parlant amb persones que han buscat pis, usant el sentit comú).

Descomposició de problemes complexos: El problema de recomanar habitatges és complex perquè involucra molts factors interrelacionats. La clau ha estat descompondre-lo en subproblemes més simples que es resolen seqüencialment. Aquesta habilitat és transferible a molts altres problemes d'IA i d'enginyeria en general.

Valor de la transparència: En aplicacions que afecten decisions importants de les persones (on viuran, quant pagaran), la transparència no és negociable. El sistema ha de poder explicar les seves decisions. Això contrasta amb molts sistemes de machine learning actuals que són "caixes negres".

Prototipatge com a metodologia: Construir versions funcionals des del principi, encara que siguin incompletes, és molt més efectiu que intentar dissenyar-ho tot perfectament abans de començar a implementar. El feedback que obtens executant el sistema t'ajuda a refinar el disseny de maneres que no hauries anticipat.

Limitacions dels sistemes simbòlics: Tot i la potència dels sistemes basats en regles, també hem vist les seves limitacions: són rígids, requereixen que tot el coneixement s'explíci manualment, no aprenen automàticament. Això ens fa valorar més els enfocaments híbrids que combinen IA simbòlica i connexionista.

En definitiva, aquesta pràctica ha estat una experiència molt enriquidora que ens ha permès aplicar conceptes teòrics a un problema real, enfrontar-nos a decisions de disseny no trivials, i desenvolupar un sistema que, amb totes les seves limitacions, realment funciona i genera recomanacions sensates. Hem après tant dels èxits com dels errors, i això és precisament el que fa que una pràctica sigui valuosa des del punt de vista formatiu.