

# Intel·ligència artificial:

## - Sistemes basats en el coneixement -

Delicado Alcántara, Luis  
Conejo Micó, Xavier  
Sanchez Ferreres, Josep

7 de desembre de 2014

# Índex

<b>1</b>	<b>Anàlisi del problema</b>	<b>2</b>
1.1	Identificació . . . . .	2
1.1.1	Definició formal de solució . . . . .	2
1.1.2	Justificació de la complexitat del problema . . . . .	3
1.1.3	Identificació de les fonts del coneixement . . . . .	3
1.2	Conceptualització . . . . .	4
1.2.1	Coneixement estàtic: La ontologia . . . . .	4
1.2.2	Procés de resolució: Divisió en subproblemes . . . . .	4
1.3	Formalització . . . . .	5
1.3.1	Informació de l'usuari: Classificació . . . . .	5
1.3.2	Informació de l'usuari: Preferències . . . . .	5
1.3.3	Determinar el model de ciutat . . . . .	6
1.3.4	Montar el viatge . . . . .	6
1.3.5	Identificació dels models de problema . . . . .	7
1.4	Implementació . . . . .	7
1.4.1	Implementació en CLIPS . . . . .	7
1.4.2	Implementació de l'ontologia . . . . .	10
<b>2</b>	<b>Documentació de l'ontologia</b>	<b>11</b>
2.1	Informació general . . . . .	11
2.2	Solució . . . . .	12
2.3	Representació de l'ontologia . . . . .	13
<b>3</b>	<b>Jocs de proves</b>	<b>14</b>
3.1	Un viatge romàntic de parella . . . . .	14
3.2	Un viatge de fi de curs . . . . .	14
3.3	Un viatge de ric . . . . .	15
3.4	Viatge per l'immerso . . . . .	16
3.5	Viatge per una família amb fills . . . . .	17
3.6	Viatge per una colla castellera . . . . .	18

# 1 Anàlisi del problema

Aquesta pràctica consisteix en resoldre un problema donat mitjançant un sistema bastat en el coneixement (*SBC*). És primordial a l'hora de dissenyar un *SBC* el fet de realitzar un anàlisi complet del domini del problema des del punt de vista de l'enginyeria del coneixement, ja que usualment, les metodologies clàssiques provinents de l'enginyeria del software no són del tot adequades per a aquesta tasca. Per a resoldre aquest problema hem seguit una versió simplificada d'un d'aquests esquema d'anàlisi i resolució del problema repartit en 5 parts: Identificació, Conceptualització, Formalització, Implementació i Prova, tal com es proposava a l'enunciat de la pràctica.

Als apartats que venen a continuació s'expliquen les decisions que hem pres a l'hora de resoldre el problema.

## 1.1 Identificació

El primer punt a tractar a l'hora de resoldre qualsevol problema és una correcta identificació d'aquest, prenent les consideracions necessàries, acotant el domini i definint correctament quin objectiu es vol assolir.

Pel nostre cas, l'objectiu a assolir és la creació d'un *SBC* que serveixi per a recomanar possibles viatges als clients d'una agència de viatges, donades les característiques dels clients, les seves restriccions i/o preferències. Tots els conceptes que hem escollit representar i una justificació pertinent es troben en els apartats de Conceptualització i Formalització.

### 1.1.1 Definició formal de solució

Definim més formalment una solució al problema com una llista ordenada de tuples  $\langle \text{Ciutat}, \text{Dies} \rangle$  on, per cada ciutat, es guarda una llista de llocs a visitar i un allotjament d'aquesta ciutat. A més, també cal donar, per cada dos ciutats en el recorregut quin mitjà de transport es farà servir.

Adicionalment s'ha de donar el preu<sup>1</sup> total del viatge a partir dels preus de totes les coses a realitzar. Tal com es definia a l'enunciat, la resposta del *SBC* ha de ser dues solucions diferents, és a dir, dos recomanacions possibles del viatge.

Definim a continuació alguns dels conceptes emprats en la definició:

**Visita:** Definim una *visita* (o *activitat*) com un a tupla  $\langle \text{Nom}, \text{Preu}, \text{Durada} \rangle$ .

Per cada ciutat la suma de les durades de les activitats no poden superar el tems d'estada a la ciutat. Per exemple ("Visita al London Eye Londres, durada mig dia, preu 25€").

---

<sup>1</sup>Tots els preus calculats són per persona. Hem considerat pertinent fer aquesta simplificació per no haver de tenir en compte els descomptes per grups grans a l'hora de generar la solució.

**Allotjament:** Definim l'*allotjament* anàlogament a la visita, però tenint en compte que la durada d'un allotjament coincideix amb el temps d'estada a la ciutat. Per exemple ("Hotel Juan Carlos I *Barcelona*, preu per dia 150€").

**Mitjà de transport:** Un mitjà de transport es defineix entre un parell de ciutats, amb un nom i un identificador. També té un preu. Per exemple ("Vol: Paris-Londres, preu 200€").

### 1.1.2 Justificació de la complexitat del problema

No costa gaire convencer-se que la complexitat del problema és bastant superior a qualsevol resoluble mitjançant cerca heurística o local.

Veiem que d'entrada el criteri –o criteris– a optimitzar no són del tot evidents i variaran en funció del tipus de client. Per exemple, hi pot haver un client ric que vulgui un viatge de luxe sense importar el preu i també un client de classe mitja/baixa amb una clara restricció de preu. Podem repetir aquest argument pràcticament per tots els punts a considerar a l'hora de fer la solució. Resulta evident que condensar tota aquesta capacitat de decisió en una funció heurística resulta massa complicat com per intentar abordar el problema amb mètodes de cerca. Pensant en això també ens adonem que és primordial el coneixement d'un expert en el domini a l'hora de resoldre el problema, ja que en el tema de les recomanacions no existeix la optimalitat objectiva, i per tant la qualitat de les solucions dependrà del coneixement d'aquest expert i de com bo sigui recomanant viatges per tal de transferir el seu coneixement a l'*SBC*.

Per tots aquests motius es pot justificar que un *SBC* és una opció adequada a l'hora de resoldre el problema.

### 1.1.3 Identificació de les fonts del coneixement

Tenint clar que cal resoldre el problema dissenyant un *SBC*, el concepte de coneixement es torna primordial. En concret en aquest apartat justifiquem les fonts de coneixement del domini que ens permetran dissenyar el sistema.

D'entrada, veiem clar que no disposem d'un expert en el domini del qual extreure el coneixement, és per això que haurem d'actuar nosaltres mateixos com experts en el tema fent servir el sentit comú a l'hora de prendre les decisions. No obstant, això no vol dir que no ens basem en coneixement real del domini, ja que els viatges són un domini on tothom té cert coneixement: ja sigui per experiència pròpia o adquirida.

En un escenari més realista per una aplicació real identificariem clarament com a fonts de coneixement els empleats de l'agència de viatges, sobretot els que tinguin contacte directe amb el client, ja que són els que

fan justament la feina de l'SBC. Per una altra banda també tenim el coneixement estàtic del món real (ciutats, hotels, etc.) que representarem a l'ontologia del domini.

## 1.2 Conceptualització

Un cop ben definit quin és el problema a resoldre, cal plantejar-se quins són els conceptes importants del domini per tal de representar-los bé com a regles o bé com a coneixement estàtic a l'ontologia. També cal plantejar-se la divisió del problema gran en subproblemes menors que donin lloc a mòduls concrets d'una implementació posterior.

### 1.2.1 Coneixement estàtic: La ontologia

Un bon punt de partida és la solució del problema. D'allà ja veiem que els conceptes de ciutat, allotjament, mitjà de transport i activitat són necessaris per a l'SBC. Aquests conceptes representen el coneixement estàtic del domini del que es disposa sense tenir en compte l'expert. Donada aquesta naturalesa de fets estàtics aquests conceptes seràn els que formaràn part de l'ontologia. Considerem que sense haver d'entrar en detalls d'implementació, la definició d'aquests conceptes ja s'ha fet prou detalladament a l'apartat anterior.

### 1.2.2 Procés de resolució: Divisió en subproblemes

Respecte el problema, hem identificat tres parts ben diferenciades: Determinar l'informació de l'usuari, raonar sobre aquella informació per tal de definir quins tipus de viatge són adequats per ell i montar el viatge final. Més concretament definim aquests problemes com:

**Obtenir informació de l'usuari:** L'objectiu d'aquest problema és ben clar:

Fer totes les preguntes necessàries a l'usuari per tal de classificar-lo en diversos tipus de perfil (nivell cultural, preferències, restriccions...) tot definint l'arquetip en el que col·loquem el client. És la part del problema que inclou una interacció directa amb l'usuari.

**Determinar el model de ciutat** Un cop sabem com és l'usuari, el següent pas és definir quins tipus de ciutats recomanarem a l'usuari, és a dir, construir un arquetip e solució.

**Montar el viatge** L'últim pas és, un cop es té el model abstracte de solució, concretar-lo en un conjunt de ciutats, allotjaments i activitats en el format explicat a la definició de la solució.

En els apartats 1.3 i 1.4 es concreten els aspectes formals i tècnics d'aquesta subdivisió de problemes.

### 1.3 Formalització

Tenint ja clars els conceptes que intervenen en el domini i sabent com s'ha subdividit el problema en subproblemes menors és el moment de formalitzar aquest coneixement. Cal, per una banda definir una ontologia<sup>2</sup> que representi el coneixement estàtic del domini com un conjunt de classes, atributs i relacions. Per altra banda és necessari definir formalment què ha de resoldre cadascun dels tres subproblemes i quines dades d'entrada i sortida tindrà cadascun d'ells per tal de facilitar l'implementació final.

Donem, a continuació, la definició formal dels subproblemes que hem utilitzat juntament amb una explicació de tots els punts pertinents. Per una definició més general de cadascun dels mòduls referir-se a la secció 1.2.2 d'aquest mateix document.

#### 1.3.1 Informació de l'usuari: Classificació

Podem veure aquest mòdul com una cadena de preguntes que pretén classificar l'usuari en unes categories prefixades. Ja es pot parlar de coneixement expert en aquest mòdul, ja que tot i que s'obté pràcticament tota l'informació de l'usuari mitjançant preguntes, no totes les preguntes són necessàries en cada cas i s'obvien en els casos pertintents.

En aquest cas hem dividit el mòdul principal en dos submòduls. El primer es dedica exclusivament a determinar el tipus de client:

**Entrada:** Aquest mòdul no té dependències d'altres mòduls i la seva única font d'informació és l'entrada de l'usuari.

**Sortida:** La sortida d'aquest mòdul es recull en tota una sèrie de camps obtinguts directament o deduïts a partir de les preguntes que es fan al client.

- *Tipus de viatge.* Possibles valors: Cultural, Diversio, Negocis, Romantic, Relax, Aventura.
- *Mitjana d'edat.* Possibles valors: Jove, Adult, Jubilats, Variada.
- *Tipus client.* Possibles valors: Grup Organitzat, Grup Amics, Institut, Família, Parella, Individual, Companys de feina.

#### 1.3.2 Obtenir informació de l'usuari: Restriccions i preferències

El segon mòdul està orientat a obtenir les preferències i/o restriccions que l'usuari vulgui imposar sobre el viatge.

---

<sup>2</sup>Veure apartat 2 pels detalls sobre la ontologia, en aquest apartat ens centrarem en la descomposició en subproblemes

**Entrada:** En aquest cas l'entrada correspon també només a l'entrada de l'usuari més alguns camps addicionals del submòdul anterior per tal de obviar algunes preguntes segons calgui.

**Sortida:** La sortida d'aquest mòdul es recull en tota una sèrie de camps obtinguts directament o deduïts a partir de les preguntes que es fan al client. Addicionalment, i com una excepció, es classifica l'usuari per nivell de vida en aquest mòdul ja que és on es pregunta pel pressupost. Seria argumentable el posar-lo en el mòdul anterior i en tot cas hem optat per aquesta opció per tal de simplificar la gestió dels mòduls.

- *Pressupost.* Dos enters, mínim i màxim d'euros.
- *Duració.* Dos enters, mínim i màxim de dies.
- *Nombre de ciutats* Dos enters, mínim i màxim de ciutats a visitar.
- *Prefereix ciutats exòtiques.* Un predicat simple, si és cert vol dir que l'usuari prefereix centrar el seu viatge en ciutats exòtiques.
- *Nivell de vida.* Possibles valors: Molt alt, Alt, Mig, Baix.

### 1.3.3 Determinar el model de ciutat

El tercer mòdul està orientat a obtenir, a partir del primer mòdul i algunes preferències de l'usuari, un model de ciutat que li interessi.

**Entrada:** En aquest cas l'entrada correspon a la descripció obtinguda sobretot en el primer mòdul juntament amb la informació que s'obté en el mòdul de restriccions i preferències.

**Sortida:** La sortida d'aquest mòdul es recull en tres camps obtinguts a partir de la informació de l'usuari.

- *Tipus de ciutat.* Pot tenir tres valors: molt coneguda, coneguda i desconeguda. Aquest camp es dedueix a partir del tipus d'activitat que s'ha obtingut en el primer mòdul juntament amb el tipus de client i si és un viatge exòtic.
- *Cualitat Hotels.* Pot tenir tres valors: alt, mitja i baix. Aquest camp depèn del nivell de vida de l'usuari i en alguns casos del pressupost i els dies que van de viatge.
- *Tipus d'activitats* Possibles valors: Cultural, Diversio, Negocis, Romàntic, Relax, Aventura.

### 1.3.4 Montar el viatge

Aquest últim mòdul té l'objectiu de, donat un arquetip de ciutat i tota l'informació recollida sobre l'usuari, montar un viatge adient per al client tot respectant les restriccions i preferències donades.

**Entrada:** L'entrada aquest cop correspon a la sortida del segon mòdul més l'informació obtinguda al mòdul de preferències.

**Sortida:** La sortida d'aquest mòdul correspon a una solució especificada a l'apartat 1.1.1.

### 1.3.5 Identificació dels models de problema

Donades aquestes definicions de problemes ens adonem com, en sí, tota l'aplicació està resolent un problema d'anàlisi mitjançant classificació heurística en 4 parts: Els dos primers mòduls realitzen les fases de recollida d'informació i d'abstracció, el tercer calcula un model de solució i per últim el quart fa un pas final de refinament per tal de convertir aquest model de solució en una solució concreta donada l'informació del domini.

Així mateix, cadascun dels problemes està resolent per si sol un problema més petit, que podem considerar d'anàlisi en tots els casos menys l'últim, on s'està realitzant una tasca clarament de síntesi.

El fet de veure l'aplicació com un agregat de subproblemes desacoplats en gran part, facilita l'última tasca d'implementació d'aquesta.

## 1.4 Implementació

En aquesta secció es detallen les diverses decisions d'implementació que hem pres a l'hora de resoldre el problema plantejat. Cal notar que una explicació més completa sobre l'ontologia del domini es troba a l'apartat 2.

Un cop detallats tots els subproblemes a resoldre cal proporcionar una implementació per a aquests. En el cas d'aquesta pràctica hem escollit un llenguatge de programació basat en regles: CLIPS, juntament amb un editor d'ontologies per a editar l'informació estàtica de l'aplicació: Protégé.

### 1.4.1 Implementació en CLIPS

Aquesta ha estat potser la part més important de la implementació de la pràctica. Per a abordar el problema hem dissenyat inicialment un proptotip senzill que feia preguntes bàsiques –preguntant a l'usuari els valors dels deftemplate– i imprimit viatges quasi aleatoris. Posteriorment hem anat iterant sobre aquest proptotip inicial afegint alguns casos que ens han semblat interessants per tal de generar un conjunt adient de jocs de proves variat.

Les ultimes iteracions del problema les hem dedicat especialment a millorar la qualitat dels viatges generats, tenint en compte cada vegada més variables com el pressupost, nombre màxim i mínim de dies i ciutats i d'altres preferències de l'usuari com poden ser el fet que vulgui visitar ciutats exòtiques.

Durant tota l'implementació hem intentat simular en molts punts el coneixement de l'expert prenent decisions anticipades a l'informació que dona



l'usuari, o presuposant coses que aquest no ha dit explícitament. Malauradament, al no ser experts en la matèria i sense el coneixement d'un expert real aquests intents de simulació en alguns casos resulten en assumpcions una mica agoserades.

### **Preguntes a l'usuari<sup>3</sup>**

Per al mòdul de preguntes a l'usuari hem seguit una estratègia de cadena de preguntes, fent èmfasi en intentar obtenir el màxim d'informació fent el mínim de preguntes per tal de no molestar en excés a l'usuari. No obstant cal notar que el cas d'ús habitual del sistema implica que cada usuari no el farà servir de manera continuada, així que el fet que les preguntes s'extenguin una mica no suposa un problema important.

En la major part dels casos s'ha optat per preguntes directes i inambigües que no facin pensar massa a l'usuari. Per exemple: “Quin nombre màxim i mínim de dies vols viatjar?”. Aquesta elecció s'ha pres per tal de facilitar la recollida d'informació.

Per tal de fer les preguntes s'han utilitzat les funcions auxiliars del mòdul `utils`, disponible a `utils.clips`.

### **Determinar el model de viatge**

Aquest mòdul simplement és un conjunt de regles senzilles, a l'estil d'implícacions de lògica de predicats que mapeja el model de client a un model de ciutat. En aquest pas és on intentem reduir el nombre de variables per tal que es pugui acotar fàcilment el model de solució.

El nombre de variables que hem escollit és de tres, les quals, es generen de diferents maneres:

**Tipus ciutat:** Aquesta variable l'obtenim a partir del tipus d'usuari, el tipus d'activitat que ha escollit, i si vol ciutats conegudes o no. La majoria de les regles tenen la mateixa estructura. En primer lloc depenent de quin tipus d'activitat ha escollit, i en algunes ocasions, depenent de quin tipus d'usuari, se seleccionen dos valors dels tres possibles. A vegades, dos tipus d'usuaris diferents amb el mateix tipus d'activitats poden tenir diferents tipus de ciutat, per exemple, si un institut vol un viatge cultural, sempre serà a ciutats molt conegudes, mentre que si és un grup d'amics, pot ser una ciutat menys coneguda.

Després d'haver eliminat un dels possibles valors, s'escull un dels dos restants depenent si vol ciutats poc conegudes o no, això vol dir que si un usuari indica que no vol ciutats conegudes les ciutats segur que

---

<sup>3</sup>L'implementació d'aquests dos mòduls es pot consultar a l'arxiu `determinar-usuari.clips`

són poc conegudes, altrament, d'entre els dos valors s'escollirà el que afavoreixi ciutats més desconegudes.

**Cualitat allotjament:** Aquesta variable l'obtenim a partir del nivell de vida de l'usuari, el numero mínim de dies i el pressupost mínim. Si l'usuari té un nivell de vida alt, vol dir que es pot permetre de tot en el viatge i per tant la qualitat de l'allotjament és alta. Per altra banda, si té un nivell de vida mitja, mirem si el seu pressupost mínim per dia és superior a una constant<sup>4</sup>, si ho és, la qualitat és alta, sinó la qualitat és mitja. Per últim, si el seu nivell de vida ens baix, tornem a mirar el pressupost mínim per dia i si és superior a 100 la qualitat, és mitja i si no, la qualitat és baixa.

**Tipus activitat:** Aquesta variable és simplement una còpia de la informació obtinguda pel mòdul anterior, ja que no és necessari, de la manera que hem pensat el problema, modificar-la.

## Montar els viatges

Aquest ha estat el mòdul el qual ha requerit més esforç a l'hora d'implementarlo. L'implementació es pot consultar a l'arxiu `montar-viatges.clips`.

Com que determinar les ciutats i els viatges entre aquestes és un problema d'una complexitat considerable (més que exponencial) obtenir una solució òptima no és factible, així que hem optat per una solució voraç.

En primer lloc l'algoritme intenta cobrir els requisits mínims de dies, ciutats i preu del viatge sense passar-se de cap dels tres escollint ciutats de manera pseudo-aleatòria dins de les ciutats de la temàtica sel·leccionada. Un cop escollides les ciutats s'escullen hotels i activitats tenint en compte l'estatus social de l'usuari. Si aquest és molt ric, s'escolliran tot tipus d'activitats diferents i de preus elevats, altrament s'escolliran activitats més barates i possiblement en menys quantitat.

Posteriorment, si ha sobrat pressupost, l'algoritme intenta fer un segon pas de refinament a la solució per tal de afegir hotels de més qualitat o més activitats, intentant no passar-se del pressupost final.

Per generar les dos solucions alhora utilitzem les pròpies regles de producció que ens ofereix clips, creant dues instàncies de solució i agafant les instàncies de solució a la precondition de cada regla.

A més, per tal d'assolir un comportament més sequencial, hem hagut d'establir la dependència lineal entre els diversos passos de construcció del viatge.

Un cop s'ha generat una solució de viatge completa s'imprimeix per pantalla amb un message handler.

---

<sup>4</sup>Hem escollit 1500 en aquest cas, caldria un estudi posterior per veure si aquest valor és prou correcte. No obstant no comptem amb dades reals a l'ontologia com per fer un anàlisi real

Cal comentar que ens ha estat impossible carregar l'ontologia en un arxiu a part ja que clips no és capaç de buscar instàncies fora del mòdul actual si no és amb un find-all-instances. Així, si creem una funció que rep una ciutat per paràmetre, els mètodes definits per a aquell paràmetre només es buscaràn dins el mòdul actual. És per això que tota l'ontologia s'inclou en el mateix fitxer que aquest mòdul.

#### **1.4.2 Implementació de l'ontologia**

L'implementació de l'ontologia ha estat pràcticament una traducció directa de les idees de l'apartat 2 al propi Protégé. La major part de les instàncies s'han omplert manualment. Cal destacar, però, el cas de les instàncies dels viatges, les quals no han estat generades manualment des de Protégé sino que han estat generades a partir d'un script en bash que emparella totes les ciutats amb vols, i assigna altres mitjans de transport de més baix pressupost. És per això que les instàncies de la classe Viatge es troben en l'ontologia del codi CLIPS però no a la ontologia en Protégé. Es pot consultar l'script utilitzat al fitxer adjunt `travel-generator.sh`

## 2 Documentació de l'ontologia

En aquesta secció explicarem com s'ha construït l'ontologia i analitzarem cadascuna de les classes que la componen. Per analitzar-les veurem per quins atributs estan compostes i quines relacions tenen amb les altres classes. També explicarem el motiu pel qual s'ha creat cadascuna de les classes.

Per explicar l'ontologia el primer que s'ha de comentar és que aquesta està dividida en dues parts diferenciades. Cal veure que encara que siguin dues parts les classes es troben altament relacionades entre elles.

### 2.1 Informació general

Aquesta és una part constant que conté la informació que proporciona l'expert i que no té res a veure amb l'usuari. En aquesta part està formada per quatre classes.

**Ciutat:** Aquesta classe conte dos atributs, el nom que és un String per identificar-la i la importància, que és un símbol, que pot tenir els valors molt coneguda, coneguda i desconeguda, ja que depenent de l'usuari, aquest voldrà anar a un tipus o un altre.

Aquesta classe està relacionada amb la classe activitat i un altre amb allotjament, ambdós amb un multislots. Aquest serveix per guardar quines són les activitats i l'allotjaments que ofereix cada ciutat i així saber que si una persona va a una ciutat quines activitats pot fer i a quins hotels es pot allotjar.

La classe ciutat queda partida en cinc subclasses, que al mateix temps es divideixen en successives subclasses en alguns casos. Això ens ajuda a l'hora de crear els vols, ja que, si són d'un altra zona, el preu serà més elevat que si són de la mateixa.

**Allotjament:** Aquesta classe conte dos atributs, el nom de l'allotjament, en un String, per poder identificar-ho i un altre atribut que és el preu de l'allotjament per dia i persona, que es guarda com un enter, que ens serveix per calcular els diners que es gastarà un usuari.

Aquesta classe també té cinc subclasses que s'utilitzen per identificar la qualitat de l'hotel depenent de quantes estrelles té i així, depenent de quin tipus d'usuari demana un viatge, s'ofereix un tipus o un altre.

**Activitat:** Aquesta classe té tres atributs, el primer, com a les classes anteriors un nom per identificar l'activitat, després tenim el seu preu, com un valor enter<sup>5</sup>, i per últim la durada de l'activitat, que es guarda com un símbol amb dos valors possibles, mig dia i un dia, i així es

---

<sup>5</sup>Considerem que comptar els decimals no té massa sentit donat l'ordre de magnitud del preu dels viatges

pot saber quantes activitats com a màxim poden fer en el temps que estigui en una ciutat.

També conte unes subclasses per identificar el tipus de l'activitat i així escollir activitats al gust de l'usuari.

**Viatge:** Aquesta classe conte dos atributs, el primer, com en les dues últimes classes, és un enter que s'utilitza per guardar el preu del viatge, el segon, es guarda com un symbol que identifica el tipus de transport que utilitza aquell viatge i pot tenir quatre valors: avio, tren, autobús i vaixell.

També conte una doble relació amb la classe ciutat. Aquesta relació ens dona la informació de quin és l'origen i el destí del viatge.

## 2.2 Solució

Aquesta part de l'ontologia no representa coneixement estàtic sobre el domini pròpiament dit, sinó que més aviat conté estructures de dades que permetin al sistema expert estructurar la informació sobre les solucions que generi del problema. En aquesta part està formada per dues classes.

**Solució parcial:** Definim una solució parcial com l'etapa que passa l'usuari en una ciutat del viatge. Amb aquesta definició podem deduir que tindrà relacions amb tres classes diferents.

Primer tenim la ciutat a la qual correspon aquesta solució parcial, després l'hotel de la ciutat en el que s'allotja i per últim un conjunt d'activitats que són les que es faran.

Després tenim dos atributs, el primer és el preu que costa fer tota la solució parcial que es guarda en un enter i després tenim la durada d'aquest que es guarda en un altre enter i ens facilita el càlcul total del temps i el preu de tot el viatge.

**Solució final:** Una instància de solució final es defineix com un viatge complet i, per tant, té una relació amb la classe solució parcial. Aquesta relació és la que guarda totes les estades de l'usuari per les ciutats que visitarà juntament amb les activitats. També té una altra relació amb la classe viatge per saber com va d'una de les ciutats a la següent. Per últim, té un atribut que conté el preu final del viatge comptant els preus dels viatges i de les estades a cada ciutat.

## 2.3 Representació de l'ontologia

En aquest apartat s'inclou un a representació gràfica de l'ontologia generada. Cal notar que, per motius evidents no ens ha estat possible incloure-la en un nivell de resolució prou adequat per tal que es puguin observar tots els detalls. És per això que s'adjunta l'arxiu `ontologia-grafic.png` amb una representació en alta resolució generada pel programa Protégé.

### 3 Jocs de proves

En aquesta secció presentem diversos jocs de proves generats per tal d'evaluar el rendiment i qualitat del nostre *SBC* tot comentant els resultats obtinguts. Per cadascuna de les proves hem inclòs les condicions<sup>6</sup>, comportament esperat i resultat de la prova. Al final de cada prova s'inclouen comentaris sobre els resultats.

#### 3.1 Un viatge romàntic de parella

Per aquesta prova demanarem una recomanació de viatge d'una parella jove a l'*SBC*. Es pot consultar l'execució complerta a l'arxiu `Jocs_de_proves_viatge_romantic.txt`.

<b>Condicions de la prova</b>	Cas típic de viatge de parella de caire romàntic per diverses ciutats poc conegudes. Entre 3 i 5 dies i per menys de 800€ per persona.
<b>Comportament esperat de l'<i>SBC</i></b>	Donada la naturalesa del viatge demanat esperem que l' <i>SBC</i> esculli un conjunt de ciutats exòtiques arreu del mon. No obstant donada la forta restricció de preu és d'esperar que s'escullin activitats de baix cost i hotels més aviat barats. Cal comentar també que s'espera que les activitats siguin de caire romàntic, donat que és la naturalesa principal del viatge.
<b>Resultats de la prova</b>	

Aenean tristique fermentum mauris, a pharetra ex varius rutrum. Aliquam consequat faucibus enim eget ullamcorper. Sed sapien odio, convallis et risus ac, condimentum tincidunt diam. Phasellus rutrum ullamcorper sem, id finibus lectus maximus sed. Nunc non ornare est, quis sodales ligula. Duis laoreet nisl vel magna auctor imperdiet. Maecenas ac magna vitae enim facilisis pharetra nec nec magna.

#### 3.2 Un viatge de fi de curs

Per aquesta prova demanarem una recomanació de viatge de final de curs a l'*SBC*. Es pot consultar l'execució complerta a l'arxiu `Jocs_de_proves_viatge_fidecurs.txt`.

---

<sup>6</sup>Cal dir que en aquest document només s'intenta donar una idea qualitativa de quina informació es donarà a l'*SBC*. A l'inici de cada apartat es fa menció al fitxer adjunt que conté l'execució completa.

<b>Condicions de la prova</b>	Es demanarà un viatge per una classe d'institut de 25 persones. Donat el baix pressupost d'aquest tipus de viatges demanarem menys de 350€ per persona. Demanarem activitats de caire cultural. Es demanarà un viatge d'entre 4 i 6 dies, anant com a molt a dues ciutats.
<b>Comportament esperat de l'SBC</b>	Estimem que l'SBC donarà un viatge vàlid dins les restriccions demanades. Tenint en compte el pressupost esperem que s'escullin activitats i hotels de baix pressupost.
<b>Resultats de la prova</b>	

Aenean tristique fermentum mauris, a pharetra ex varius rutrum. Aliquam consequat faucibus enim eget ullamcorper. Sed sapien odio, convallis et risus ac, condimentum tincidunt diam. Phasellus rutrum ullamcorper sem, id finibus lectus maximus sed. Nunc non ornare est, quis sodales ligula. Duis laoreet nisl vel magna auctor imperdiet. Maecenas ac magna vitae enim facilisis pharetra nec nec magna.

### 3.3 Un viatge de ric

Per aquesta prova demanarem una recomanació de viatge a l'SBC per una persona de classe social molt alta i amb molts diners. Es pot consultar l'execució complerta a l'arxiu `Jocs_de_proves_viatge_ric.txt`.

<b>Condicions de la prova</b>	Es demanarà un viatge sense cap limitació pel pressupost per una única persona d'edat adulta, buscant una temàtica principalment de relax. Com a restriccions es demanarà que el viatge duri dues setmanes i es volen visitar almenys 6 ciutats
<b>Comportament esperat de l'SBC</b>	Creiem que el sistema expert donarà un viatge substancialment més car que en els casos anteriors. Donat l'elevat nombre de ciutats a l'ontologia que ofereixen una temàtica de relax s'espera que pugui generar dos viatges diferents, tot i que amb molta seguretat es repeteixi alguna ciutat. Lògicament el preu de les activitats serà el més elevat.
<b>Resultats de la prova</b>	

Aenean tristique fermentum mauris, a pharetra ex varius rutrum. Aliquam consequat faucibus enim eget ullamcorper. Sed sapien odio, convallis et risus ac, condimentum tincidunt diam. Phasellus rutrum ullamcorper



sem, id finibus lectus maximus sed. Nunc non ornare est, quis sodales ligula. Duis laoreet nisl vel magna auctor imperdiet. Maecenas ac magna vitae enim facilisis pharetra nec nec magna.

### 3.4 Viatge per l'imsero

Per aquesta prova demanarem una recomanació de viatge a l'*SBC* per un grup de jubilats. Es pot consultar l'execució complerta a l'arxiu `Jocs_de_proves_viatge_imsero.txt`

<b>Condicions de la prova</b>	Es demanarà a l' <i>SBC</i> un viatge per un grup de gent gran (40 persones) amb la idea de fer un tour per diverses ciutats (entre 3 i 5), en una setmana. El pressupost serà entre 200 i 300€ per persona. Es demanarà explícitament que s'intenti ajustar al màxim el pressupost.
<b>Comportament esperat de l'<i>SBC</i></b>	Creiem que l' <i>SBC</i> recomanarà principalment activitats de relax. Escollint hotels i mitjans de transport barats. És molt possible que amb un pressupost tan baix i demanant tantes ciutats no sigui capaç d'ajustar-se al pressupost màxim.

#### Resultats de la prova:

---

##### VIATGE 1

El viatge trobat es surt del pressupost màxim per: 65 euros

Preu total (persona): 365 euros

Duració del viatge: 6 dies

Ciutats: Rio de Janeiro (1 dies), Barcelona (2.0 dies),  
Montecarlo (2.0 dies), Paris (2.0 dies)

Visites: Rio de Janeiro: RJ Massagem;  
Barcelona: Spa Aire;  
Montecarlo: Le Bar Americain, Japanese Gardens;  
Paris: Mamabali, Aquamoon Spa

Allotjament: Rio de Janeiro: Copinha Hostel, Barcelona: Alberge  
Pere Tarres,

Montecarlo: Alberge Pere Tarres, Paris: Pacific Hotel

Viatges: Origen -> Barcelona: tren, Barcelona -> Montecarlo: tren,  
Montecarlo -> Paris: tren, Paris -> Origen: avio

##### VIATGE 2

El viatge trobat es surt del pressupost màxim per: 48 euros

Preu total (persona): 348 euros

Duració del viatge: 6 dies

Ciutats: Rio de Janeiro (1 dies), Barcelona (2.0 dies)  
Paris (2.0 dies), Montecarlo (2.0 dies)

Visites: Rio de Janeiro: RJ Massagem; Barcelona: Spa Aire; Paris:  
 Mamabali  
 Aquamoon Spa; Montecarlo: Le Bar Americain, Japanese Gardens  
 Allotjament: Rio de Janeiro: Copinha Hostel, Barcelona: Alberge  
 Pere Tarres,  
 Paris: Pacific Hotel, Montecarlo: Alberge Pere Tarres  
 Viatges: Origen -> Barcelona: tren, Barcelona -> Paris: tren  
 Paris -> Montecarlo: tren, Montecarlo -> Origen: tren

---

Cal destacar que l'SBC no ha estat capaç de trobar dos viatges diferents donats els requisits dels usuaris –tantes ciutats i tan barates–. No obstant això, per tal de complir la restricció interna que els viatges havien de ser diferents, observem com l'ordre en que es visiten les ciutats està desordenat.

Deixant de banda aquest fet, veiem com s'ha quasi-complert la restricció de preu en els dos casos, i que per assolir-ho s'han escollit poques activitats d'entre les més barates de cada ciutat. Respecte els allotjaments veiem com s'han escollit els de pitjor qualitat.

Veiem també com el sistema ha decidit centrar el viatge en una temàtica de relax, i les activitats s'han escollit pertinentment per tal de reflexar aquest fet.

En general, estem satisfets amb el resultat del joc de proves.

### 3.5 Viatge per una família amb fills

Per aquesta prova demanarem una recomanació de viatge a l'SBC per una família amb fills. Es pot consultar l'execució complerta a l'arxiu `Jocs_de_proves_viatge_familia.txt`.

<b>Condicions de la prova</b>	Es demanarà un viatge per una família de 5 persones. El viatge s'haurà d'enfocar principalment a l'entreteniment pels pares i no tant pels fills (preferència de l'usuari). Es disposa d'un pressupost mitjà, entre 600 i 800€ per persona. El viatge es vol enfocar únicament a una ciutat i la durada d'aquest ha de ser com a mínim de 5 dies.
<b>Comportament esperat de l'SBC</b>	Creiem que l'SBC serà capaç de complir amb els requisits de l'usuari considerant el contingut de l'ontologia. Esperem que es doni un viatge d'almenys 5 dies en una ciutat amb almenys una activitat per cada dos dies <sup>7</sup> . En no haver-hi múltiples ciutats el pressupost es podrà complir sense cap mena de problema.

#### Resultats de la prova:

---

VIATGE 1:

Preu total (persona): 358 euros  
Duració del viatge: 5.0 dies  
Ciutats: Florencia (5.0 dies)  
Visites: Florencia: Cinque Terre  
Allotjament: Florencia: Mia Cara  
Viatges: Origen -> Florencia: avio, Florencia -> Origen: tren

VIATGE 2:  
Preu total (persona): 384 euros  
Duració del viatge: 5.0 dies  
Ciutats: Toronto (5.0 dies)  
Visites: Toronto: Edge Walk at the CN Tower, Advanced Approaches to  
          Massage,  
                                Edge Walk at the CN Tower  
Allotjament: Toronto: Bond Place Hotel  
Viatges: Origen -> Toronto: tren, Toronto -> Origen: tren

---

Veiem que el pressupost s'ha complert, no obstant això el nostre *SBC* ha actuat de manera excessivament conservadora: S'han escollit hotels i activitats massa barats. Caldria fer un estudi en profunditat de quins són els valors òptims per establir els diversos nivells de vida dels usuaris, i totes aquestes dades són coneixement expert (del qual no disposem). Tot i això el viatge segueix el comportament que esperavem.

Es pot observar, que, malauradament s'ha repetit una activitat. Associem aquest problema a un error en la ontologia –l'activitat es trobava definida dues vegades a la llista d'activitats– I per tant no a un mal funcionament de l'SBC.

Considerem que el resultat d'aquest joc de proves manifesta una manca del nostre *SBC* que es podria arreglar amb posterior anàlisi del domini i programant regles més realistes en el sistema.

### 3.6 Viatge per una colla castellera

Per aquesta prova demanarem una recomanació de viatge a l'*SBC* per als membres d'una colla castellera. Es pot consultar l'execució completa a l'arxiu `Jocs_de_proves_viatge_castellers.txt`

<b>Condicions de la prova</b>	Es demanarà un viatge per un grup nombrós (30 persones) per una colla castellera. Si el sistema pregunta direm que hi ha nens al grup. El viatge demanat haurà de durar almenys 2 dies, cada dia a una ciutat diferent i per menys de 150€ per persona.
<b>Comportament esperat de l'SBC</b>	Creiem que el sistema complirà totes les restriccions menys possiblement la de preu, la qual es passi una mica de pressupost donat que s'està requerint un viatge a dues ciutats diferents per molt poc preu. Suposem que les dues ciutats escollides seràn prou properes i probablement no s'esculli avió com a mitjà de transport.

### Resultats de la prova:

---

#### VIATGE 1:

El viatge trobat es surt del pressupost màxim per: 116 euros  
Preu total (persona): 266 euros  
Duració del viatge: 2.0 dies  
Ciutats: Sidney (1.0 dies), Munich (1.0 dies)  
Visites: Sidney: Ocean Extreme;  
Munich: Heliflieger, DAV Kletter  
Allotjament: Sidney: Highfield Private Hotel,  
Munich: Hotel Modern  
Viatges: Origen -> Sidney: tren Sidney -> Munich: tren  
Munich -> Origen: tren

#### VIATGE2:

El viatge trobat es surt del pressupost màxim per: 38 euros  
Preu total (persona): 188 euros  
Duració del viatge: 2.0 dies  
Ciutats: Chicago (1.0 dies), Sidney (1.0 dies)  
Visites: Chicago: Kayak Chicago, Sidney: Ocean Extreme  
Allotjament: Chicago: Chinatown Hotel, Sidney: Highfield Private Hotel  
Viatges: Origen -> Chicago: tren, Chicago -> Sidney: tren  
Sidney -> Origen: tren

---

Tal com sospitavem, el sistema no ha pogut complir el pressupost mínim i així ens ho indica. Respecte les demés coses, ha complert en els dos casos la durada del viatge, escollint els hotels de més baixa qualitat. Respecte a les activitats podem veure com també s'han escollit les més barates.

La temàtica del viatge és clarament de diversió, fet que s'ajusta perfectament als potencials gustos d'un grup com podria ser una colla castellera –Almenys segons el coneixement nostre *SBC*–

En general quedem satisfets amb la resposta de l'*SBC* i podem dir que és el que ens esperavem.