

Práctica 2

Joan Hervás, Xènia Calisalvo, Jaume Baqueró

Mayo 2022

Índice

1. Introducción	4
2. Identificación del problema	4
2.1. Descripción del problema	4
2.2. Viabilidad de la solución	4
2.3. Fuentes de conocimiento	4
2.4. Objetivos	5
3. Conceptualización	5
3.1. Conceptos del dominio	5
3.1.1. Ciudad	5
3.1.2. Transporte	6
3.1.3. Alojamientos	6
3.1.4. Punto de Interés	6
3.1.5. Tipo Objetivo	6
3.2. Conceptos del problema	6
3.2.1. El Cliente	7
3.2.2. El Viaje	7
3.2.3. Tiempo por ciudad	7
3.2.4. Puntuación de ciudad/punto de interés	7
3.2.5. La solución	8
3.3. Subproblemas de la resolución	8
3.3.1. Recogida de la información personal	8
3.3.2. Filtrado	8
3.3.3. Análisis de los datos del cliente	8
3.3.4. Construcción de la solución	8
3.3.5. Impresión de la solución	8
3.4. Ejemplos del conocimiento experto	8
3.4.1. Flujo de razonamiento	9
4. Formalización	10
4.1. Ontología	10
4.1.1. Determinar el dominio y la cobertura de la ontología	10
4.1.2. Definir las clases y su jerarquía	10
4.1.3. Definir las relaciones entre clases	11
4.1.4. Definir las instancias de las clases	11
4.2. Método de Resolución	12
4.2.1. Restricciones globales	12
4.2.2. Abstracción de datos y asociación heurística	12
4.2.3. Adaptación	12
4.2.4. Conjunto de operadores	12
4.2.5. Restricciones de elección de operadores	12
5. Implementación	13
5.1. Implementación de la ontología	13
5.2. Módulos	13
5.2.1. Módulo MAIN	13
5.2.2. Módulo preguntas	13
5.2.3. Módulo filtro	14
5.2.4. Módulo construcción	15
5.2.5. Módulo print solución	15

5.2.6.	Módulo lógica segunda solución	16
5.2.7.	Módulo print segunda solución	16
6.	Juegos de prueba	16
6.1.	Prueba 1	16
6.1.1.	Descripción del caso	16
6.1.2.	Solución esperada	16
6.1.3.	Resultado obtenido	16
6.2.	Prueba 2	17
6.2.1.	Descripción del caso	17
6.2.2.	Solución esperada	17
6.2.3.	Resultado obtenido	17
6.3.	Prueba 3	18
6.3.1.	Descripción del caso	18
6.3.2.	Solución esperada	19
6.3.3.	Resultado obtenido	19
6.4.	Prueba 4	20
6.4.1.	Descripción del caso	20
6.4.2.	Solución esperada	20
6.4.3.	Resultado obtenido	20
6.5.	Prueba 5	20
6.5.1.	Descripción del caso	20
6.5.2.	Solución esperada	21
6.5.3.	Resultado obtenido	21
7.	Conclusiones	22

1. Introducción

El objetivo de esta práctica es resolver un problema mediante la construcción de un Sistema Basado en el Conocimiento (SBC), con la ayuda del lenguaje CLIPS. Para ello diseñaremos una ontología utilizando Protegè para luego exportarla a CLIPS y diseñar un conjunto de módulos y reglas que utilicen esta ontología para razonar. Finalmente diseñaremos unos juegos de prueba y los ejecutaremos para comprobar que el sistema funciona correctamente.

2. Identificación del problema

2.1. Descripción del problema

Este proyecto consiste en la creación de viajes a medida para los clientes. Concretamente este tiene que ser capaz de a partir de unas preguntas a los usuarios y la información recopilada a partir de estas para poder realizar dos soluciones compuestas por las ciudades a visitar, el hospedaje de estas ciudades, los puntos de interés de estas, los días a quedarse, etc. Es decir, básicamente tiene que ser capaz de devolver todo aquello que compone un viaje.

Para esto pediremos un conjunto de información a nuestros clientes como la disponibilidad temporal de estos, máximo y mínimo de días para el viaje, presupuesto a gastar, preferencias en cuanto al tipo objetivo del viaje, si es por un evento especial y el tipo del viaje. Por ejemplo alguien puede preferir un viaje cultural compuesto por visitas a museos, templos y demás, y otra persona puede preferir un viaje de descanso donde pasar el tiempo en unas buenas aguas termales. Esto suele venir determinado por factores como lo que le gusta a cada uno pero también por si será un viaje familiar, uno de pareja o uno en solitario así que toda esta información será parte de la definición del problema.

2.2. Viabilidad de la solución

En este punto vamos a discutir diferentes métodos de resolución y porque en este caso la mejor opción es optar por un SBC. Para ello vamos a comentar otros métodos vistos en clase.

El problema descrito anteriormente consiste en encontrar una combinación de ciudades que se adapte de la mejor forma posible al cliente, teniendo en cuenta las restricciones y preferencias que propone. Para encontrar la combinación más adecuada de ciudades deberíamos recorrer todo el espacio de ciudades, para obtener las que cumplieran las restricciones y preferencias del usuario. Solucionar el problema como un problema de búsqueda sería demasiado complejo, ya que la heurística que marca las restricciones y preferencias tiene demasiadas variables.

Al tener conocimiento previo de las preferencias del usuario y la información de la ciudades, podemos resolver el problema mediante un Sistema Basado en el Conocimiento. Al construir un SBC específico para el problema, podremos encontrar una solución para distintos tipos de usuarios y viajes, lo cual nos permite concluir que construir un SBC es viable para resolver este problema.

2.3. Fuentes de conocimiento

Las fuentes de conocimiento se definen como aquellos recursos sobre los cuales nuestro sistema de SBC se basa y a partir de lo cual toma las decisiones correspondientes. Con esta definición podemos decir que para nuestro caso particular, necesitamos información sobre Ciudades y sus atributos relevantes como destinación de un viaje, e información de los lugares que están relacionados con estas Ciudades (entre ellos Puntos de Interés y Alojamientos).

Hemos tenido que construir una base de datos que contiene datos relevantes para cualquier solución que componga el viaje. Ejemplos muy claros podrían ser los nombres de cada ciudad, o los medios para ir de una ciudad a otra, a demás de sus precios. También contamos con otras estructuras menos "tangibles", como tipos de objetivo de viaje, ya que para dar una buena solución necesitamos poder representar algo tan abstracto como la afinidad entre un usuario y una ciudad, por ejemplo.

Aún así hay un límite para aquello que nuestro sistema puede saber a priori. No hay manera de tener datos sobre nuestro cliente sin que nos lo comunique él mismo.

Para solucionar este problema, contamos con un módulo que pregunta al usuario sus preferencias y limitaciones, y de éste modo nuestro sistema puede obtener toda la información necesaria, es decir, la información del cliente.

De todos modos no es completamente intuitivo saber qué tipo de información es necesaria para poder cuantificar afinidad, de modo que hemos buscado por internet qué puede ser relevante de cara a planificar un viaje, a demás de nuestro propio criterio como usuarios de éste tipo de servicios.

De ésta búsqueda hemos observado que los viajes pueden tener distintos motivos (diversión o cultura, por ejemplo), y que los lugares dentro de una ciudad que se van a visitar pueden variar en función de si se viaja en familia, con niños, o en otros formatos. También hemos tenido en cuenta el hecho de si el viaje era con el motivo de algún evento especial y le hemos dado relevancia a algunos lugares a partir de esta información, por ejemplo un viaje de bodas suele ser un viaje romántico y por ende vamos a incrementar la relevancia de los puntos románticos.

Con todo esto en cuenta, vimos que necesitábamos un modo para discernir ciudades (o puntos de interés) más o menos relevantes, así que les asignamos una puntuación, que sacamos a partir de información de expertos en este caso fuentes de internet o nuestras propias experiencias, que incrementa en función de cuántos atributos tiene en común con los gustos obtenidos del cliente.

También consideramos relevante otros factores como los gustos de lujo, o el desagrado hacia ciertos tipos de transporte (miedo a volar, marearse en barco, etc).

2.4. Objetivos

Nuestro sistema debería cumplir los siguientes objetivos para poder resolver cualquier situación con éxito:

- Interactuar con el cliente para obtener toda la información necesaria respecto a sus preferencias y restricciones sobre el viaje.
- Aprovechar todo lo posible la información obtenida de los usuarios y la extraída de la base de instancias que hemos creado.
- Ordenar las paradas del viaje y las actividades a realizar en cada una de según cuáles cumplen mejor las restricciones y preferencias propuestas por nuestro cliente.
- Establecer una secuencia lógica entre las paradas del viaje.
- Presentar al usuario el viaje final obtenido por el sistema junto a las restricciones que se cumplen y los problemas encontrados

3. Conceptualización

El objetivo es recomendar dos viajes distintos que cumplan lo máximo posible las restricciones y preferencias del usuario. Nuestro dominio se ciñe a las ciudades, los transportes entre ellas, sus puntos de interés y de qué tipo es cada ciudad y punto de interés.

3.1. Conceptos del dominio

3.1.1. Ciudad

Lo primero que debemos conocer son las distintas piezas de construcción que usaremos para forjar el viaje, es decir las ciudades. Nuestras ciudades contienen los siguientes atributos:

- **Nombre:** Para poder identificar las ciudades.

- **Puntos de interés:** Un conjunto de relaciones que indican los puntos de interés que tiene la ciudad.
- **Alojamientos:** Un conjunto de relaciones que indican los alojamientos que tiene la ciudad.
- **Continente:** Marca a qué continente pertenece la ciudad.
- **Puntuación:** Puntuación base de la ciudad.
- **Tipos:** Un conjunto de relaciones con tipos de objetivo para poder potenciar la puntuación según las preferencias del cliente.

3.1.2. Transporte

Para unir las ciudades en el viaje, necesitamos transportes. Nuestros transportes pueden ser de cuatro tipos: Avión, Tren, Autobús o Barco. Cada transporte guarda la siguiente información:

- **Origen:** Referencia a la ciudad de origen del transporte.
- **Destino:** Referencia a la ciudad de destino del transporte.
- **Precio:** Precio del transporte. Este precio variará dependiendo de la distancia entre origen y destino y del tipo de transporte.

3.1.3. Alojamientos

Cada ciudad tiene un conjunto de alojamientos, los cuales guardan información sobre lo siguiente:

- **Calidad:** Número de estrellas del alojamiento. En caso de ser 0, se trata de un hostel o albergue.
- **Precio por noche:** El precio por noche del alojamiento. No tiene por qué corresponderse con la calidad, aunque ésta influye.
- **Ciudad a la que pertenece:** Referencia a la instancia de la ciudad a la que pertenece el alojamiento.

3.1.4. Punto de Interés

Cada ciudad contiene también un conjunto de puntos de interés, que tienen información sobre su precio, su tipo que se corresponde con los tipos objetivo como cultural o gastronómico, a qué ciudad pertenecen y una puntuación base asignada con conocimiento de experto.

- **Precio:** Cuánto cuesta visitar el punto de interés.
- **Tipo:** Una conjunto de referencias a los tipo objetivo del punto de interés.
- **Ciudad:** Una referencia a la instancia de la ciudad a la que pertenece el punto de interés.
- **Puntuación:** La puntuación base asignada al punto de interés.

3.1.5. Tipo Objetivo

Tanto ciudades como puntos de interés tienen asociados tipos de objetivo, que son los diferentes objetivos que puede buscar el cliente en su viaje. Esto sirve para poder potenciar las puntuaciones de las ciudades y puntos de forma que se le de preferencia a las ciudades y puntos de interés que mas se correspondan con la búsqueda del usuario y así poderlas ordenar para encontrar la mejor solución

3.2. Conceptos del problema

Nuestro problema consta de conceptos como un cliente, que tiene sus preferencias (como por ejemplo Ciudades románticas), y sus limitaciones (como por ejemplo su presupuesto, o los días que quiere viajar). Las limitaciones nos permiten definir una solución, es decir, son requisitos imprescindibles a cumplir, en cambio las preferencias solo harán que se le de mas peso a algunos puntos por encima de otros. A demás del cliente, el otro concepto principal es el viaje, que tiene también sus parámetros relevantes.

3.2.1. El Cliente

Como hemos mencionado previamente, el cliente tiene información que inserta muchas partes móviles en el problema. Procedemos a hablar sobre ellas.

- **Presupuesto:** Intuitivamente, el presupuesto de nuestro cliente. Necesario para saber cuánto se puede invertir en el viaje.
- **Sacrificio de calidad:** La preferencia del cliente según si quiere ceñirse al presupuesto, o si se puede invertir un poco más de cara a proporcionar mejores resultados.
- **Avión/Tren/Bus/Barco:** Parámetros relevantes para saber si el cliente prefiere o no usar dichos transportes. En caso negativo, nunca se sugerirán transportes de dicho tipo.
- **Origen:** La ciudad de origen del cliente. Necesario para saber de dónde partir y a donde volver al final.
- **Evento:** Información sobre un evento especial para el viaje (Boda o fin de curso por ejemplo).
- **Lujo:** Preferencia del cliente para la calidad de los hoteles. En caso afirmativo, sólo se sugerirán alojamientos de calidad 4* o más.
- **Objetivos:** Motivos del viaje. Opciones pueden ser diversión o gastronomía. Puede haber más de uno.
- **Formato del grupo:** De cara al problema es relevante saber si se viaja solo, en pareja u otros formatos.

3.2.2. El Viaje

Para el formato de viaje también tenemos datos necesarios, como los días que se quiere viajar, o el continente que se quiere visitar.

- **Días de viaje:** Los días que va a durar el viaje.
- **Número de ciudades:** Se preguntan al cliente la cantidad de ciudades que quiere visitar.
- **Continente:** El continente que quiere visitar el cliente. Se proporcionarán sólo ciudades de ese continente siempre que se pueda.

3.2.3. Tiempo por ciudad

Para definir el tiempo que un cliente se va a hospedar en una ciudad decidimos que la mejor opción era que todas las ciudad se repartieran un tiempo equitativa. Esto es debido a que al no tener mas información sobre si un cliente prefiere mas días en una ciudad que otra era mejor optar por esta opción. A partir de aquí le preguntábamos al cliente el intervalo de días en los que estaría dispuesto a viajar, a continuación hacíamos la media entre estos días y finalmente aproximábamos al numero de días en ese intervalo mas cercano al un divisible de días entre numero de ciudades. Esto es por el motivo explicado arriba de que queríamos repartir equitativamente los días y de esta forma conseguimos una forma elegante de solventar el problema cumpliendo los requisitos del cliente.

3.2.4. Puntuación de ciudad/punto de interés

Aun qué cada ciudad/punto de interés tiene una puntuación fija dada según un experto hay otra puntuación, internamente es la misma solo que varia en tiempo de ejecución, que esta es en función de las preferencias del usuario. Para calcularla simplemente iremos duplicando el valor de puntuación inicial por cada "match" que tengamos con los requerimientos del cliente.

3.2.5. La solución

Por último, la solución que nuestro sistema proporciona se define como una lista de lo que nosotros llamamos 'Estancias', a demás de un precio final.

- **Estancia:** Contiene información sobre qué ciudad se visita, cuantos días, qué alojamiento se usa, qué lugares se visitarán, y el precio del viaje en particular.
- **Precio final:** El precio acumulado en todas las estancias y los viajes entre ciudad y ciudad, incluyendo el último vuelo de vuelta.

3.3. Subproblemas de la resolución

Nuestro problema se puede dividir fácilmente en un conjunto de subproblemas, ya que como se ha descrito en los objetivos el sistema debe hacer más de una actividad. Hemos dividido la tarea a realizar en los siguientes subproblemas.

3.3.1. Recogida de la información personal

El primer subproblema consiste en obtener del cliente la información sobre sus preferencias y restricciones mediante preguntas de todo tipo. Son preguntas extensivas así que la cantidad de preguntas es grande, pero lo hemos diseñado de manera que algunas preguntas se obvien en función de las respuestas a preguntas previas.

3.3.2. Filtrado

Con la información obtenida del cliente, descartamos las ciudades y/o transportes que no cumplan sus restricciones o limitaciones.

3.3.3. Análisis de los datos del cliente

Una vez obtenida la información y filtrados los viajes disponibles, debemos analizar los datos obtenidos y aumentar las puntuaciones de las diferentes ciudades y puntos de interés que cumplan mejor las preferencias de nuestro cliente.

3.3.4. Construcción de la solución

En este subproblema se ordenan las ciudades por la puntuación que han obtenido, de mayor a menor, y se van añadiendo a la solución construyendo estancias, en las que se escogen alojamientos y puntos de interés también ordenados de mayor a menor, teniendo en cuenta siempre que no nos pasemos del presupuesto acordado. Como en nuestro enunciado se piden dos soluciones este apartado también se hará dos veces.

3.3.5. Impresión de la solución

Por último, una vez construida la solución, debemos mostrarla al cliente, explicando el viaje y las preferencias que se han cumplido, así como los problemas que hayan sucedido.

3.4. Ejemplos del conocimiento experto

Algunos ejemplos de conocimiento experto pueden ser:

- Un cliente nunca va a estar menos días que los días mínimos ni mas días que los días máximos proporcionados.
- Un cliente siempre va a preferir aquellas ciudades que se correspondan con el tipo objetivo que han pedido para el viaje.

- Igual que en el punto anterior pero para los puntos de interés.
- El cliente preferirá que el viaje este en un mismo continente ya que sino el tiempo en transporte puede ser mayor que el tiempo en el destino
- Un cliente no querrá sobrepasar su presupuesto a no ser que este previamente indicado por el mismo y aun así este presupuesto nunca sera superado en mas de un 20 por ciento ya que sino este dejara de tener sentido.
- Un cliente que no haya seleccionado viaje de lujo como opción siempre preferirá el hotel mas barato posible
- Un cliente que si haya seleccionado viaje de lujo como opción siempre preferirá el hotel con mas calidad posible.
- Un cliente que no quiera utilizar un transporte sera por motivos de fuerza mayor y por tanto nunca se va a incluir en una solución una opción con ese transporte
- Un cliente que tenga como motivo especial del viaje una boda querrá un viaje romántico
- Un cliente que viaje por motivo de un cumpleaños querrá experiencias y diversión
- Un cliente que viaje por motivo de un viaje de fin de curso equivaldrá a un viaje con amigos

3.4.1. Flujo de razonamiento

Para todos estos razonamientos hemos utilizado una combinación entre nuestras propias experiencias y una búsqueda exhaustiva en internet. En casos como el tipo objetivo de un punto de interés o una ciudad hemos buscado en internet cuales son las principales atenciones turísticas y que tipo de gente suele recomendarte el lugar. En cambio en factores como determinar que lugares preferirá que visitar una persona que viaja sola hemos usado una combinación entre ambas, inicialmente buscamos en internet para ver que nos recomendaba y a partir de aquí pensamos que lugares nos imaginábamos como posibles puntos. Hemos usado el mismo método para viajes con familia, donde por ejemplo decidimos que Disneyland era una buena opción, o en viajes de amigos, donde lo que se busca es diversión y experiencias.

Para cumplir con todo esto primero hemos realizado las preguntas que nos daban la información del usuario. A continuación hemos eliminado todas aquellas instancias (o marcado para que no se cojan) que no queremos en nuestra solución. Finalmente hemos actualizado las puntuaciones de cada lugar con las preferencias y ordenado a partir de aquí para que se cojan en orden.

4. Formalización

En esta sección explicaremos el proceso de construcción de nuestra ontología, la división de subproblemas y justificaremos todas las decisiones tomadas.

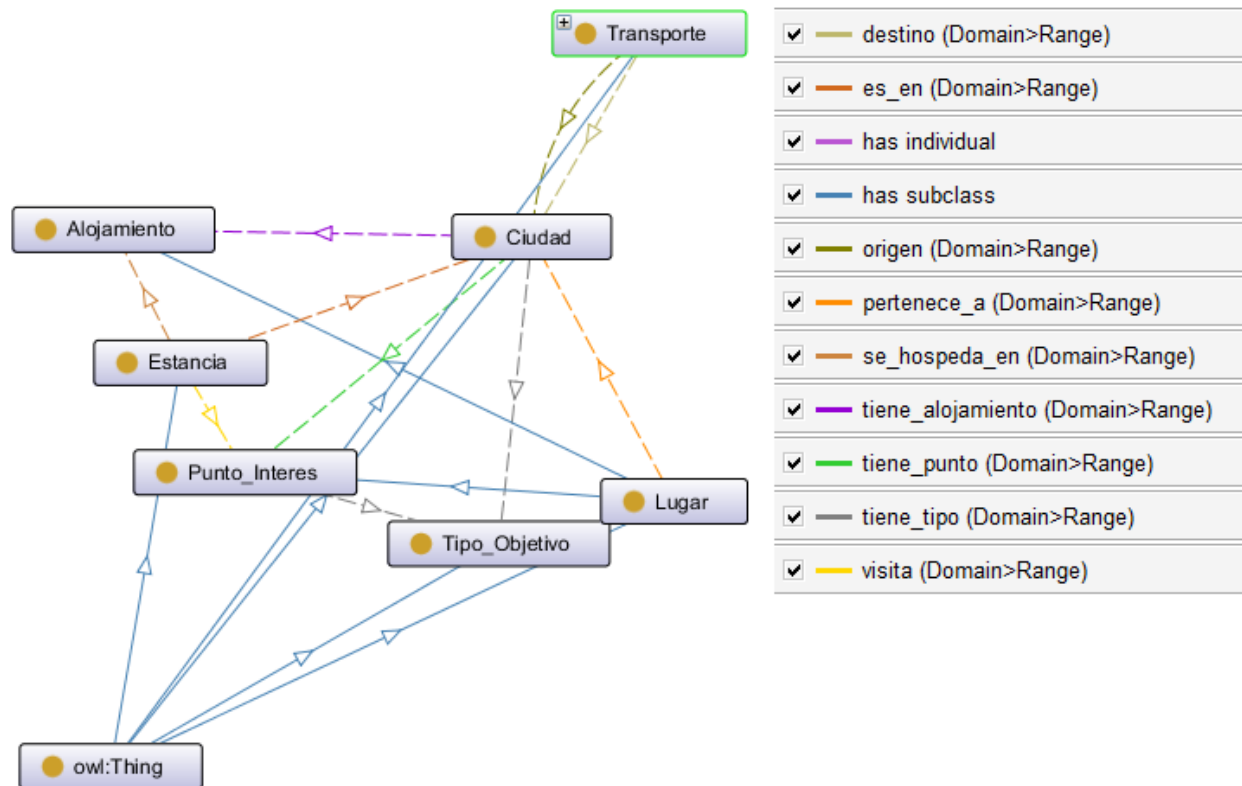
4.1. Ontología

4.1.1. Determinar el dominio y la cobertura de la ontología

En el apartado de conceptualización hemos hablado de todos los conceptos que forman nuestro sistema. De cara a crear la ontología simplemente los trasladamos a un formato más formal. Los puntos más claves de la ontología son las ciudades y los transportes, pues son el esqueleto de lo que define el problema. Los alojamientos, puntos de interés y atributos de afinidad existen para poder dar una decisión informada.

4.1.2. Definir las clases y su jerarquía

Nuestra ontología queda de la siguiente manera:



Se puede observar como en éste grafo falta la parte relacionada con la información relacionada con el usuario. Esto se debe a que esta información no forma parte de una clase en sí, si no que nuestro sistema lo almacena todo en respectivas variables globales para el sencillo acceso e interpretación.

Al principio del desarrollo de la ontología, estas clases existían, pero según más progresábamos, más nos dimos cuenta de que nos resultaría más fácil hacerlo del otro modo.

Esto significa que, si bien no existen visiblemente en la ontología, sí están presentes en el sistema, aunque de un modo más abstracto.

4.1.3. Definir las relaciones entre clases

Las relaciones entre nuestras clases existen para representar algunos de los atributos comentados en la conceptualización. Ya que algunos atributos son valores con tipos comunes (números o strings), pero algunos requieren como valores una o más instancias de otras clases. De este modo ha quedado como lo siguiente:

- **Ciudad:** Se relaciona hacia **Alojamiento, Punto de Interés y Tipo de Objetivo**, ya que queremos ser capaces de, desde una ciudad, poder acceder a todos los datos sobre cualquiera de estas instancias.
- **Estancia:** Se relaciona hacia **Alojamiento, Ciudad y Punto de Interés**. Estas relaciones fueron necesarias porque los alojamientos y puntos de interés son un subset de los que contiene una ciudad concreta, de modo que optamos por esta solución para poder acceder de modo sencillo y rápido.
- **Transporte:** Se relaciona hacia **Ciudad** dos veces. Estas relaciones fueron necesarias para poder comunicar dos ciudades. Una relación representa el origen y otra el destino. Esta relación nos permite acceder a **todos** los datos tanto de la ciudad de origen como la de destino, lo cual ha sido extremadamente útil.
- **Punto de Interés:** Se relaciona hacia **Tipo de Objetivo**. Se usa para poderse relacionar con los distintos tipos, y permite tener más de uno. Hemos optado por esta solución para poder tener varias instancias de Tipo de Objetivo asociadas a un mismo punto de interés. Similar a las ciudades.

4.1.4. Definir las instancias de las clases

Para poder definir las instancias, hemos puesto 5 ciudades cada miembro del equipo, cada uno de un continente distinto, y de cada ciudad, hemos instanciado todo lo que hemos considerado relevante de cara al sistema.

Un ejemplo de instanciación de ciudad:

```
([Barcelona] of Ciudad
  (tiene_alojamiento [Generator_Barcelona] [HLG_CityPark_Pelayo]
    [SM_AlojamientoTeatre_Auditori] [W] [Vincci_Maritimo])
  (tiene_punto [Parc_Guell] [Mercat_Boqueria] [Sagrada_Familia] )
  (tiene_tipo [Culinario] [Cultural])
  (es_conocido "true")
  (nombre "Barcelona")
  (puntuacion 9)
  (visitado "no")
  (continente "Europa")
)
```

Un ejemplo de instanciación de Punto de interés:

```
([CN_tower] of Punto_Interes
  (tiene_tipo [Experiencias])
  (es_conocido "true")
  (puntuacion 8)
  (pertenece_a [Toronto])
  (precio 28)
```

Un ejemplo de instanciación de Alojamiento:

```
([Nunurco] of Alojamiento
  (calidad 3)
  (pertenece_a [Chachapoyas])
  (precio 19)
```

Un ejemplo de instanciación de Transporte:

```
([Nueva_York-Berlin] of Avion
(destino [Berlin])
(origen [Nueva_York])
(precio 373)
```

Y cada una de las otras instancias similarmente, con los datos requeridos por su estructura de clase.

4.2. Método de Resolución

4.2.1. Restricciones globales

En nuestro problema tenemos algunas restricciones que limitan nuestras decisiones más allá de sólo afinidad. Tenemos los **días** máximos/mínimos de viaje, **ciudades** que se quiere visitar y, por supuesto, **presupuesto**. Por decisión nuestra, también hemos añadido deliberadamente la opción de poner limitaciones sobre la calidad de los hoteles y los tipos de transporte. Aunque esto pueda crear más situaciones donde no exista una solución, pensamos que es preferible a que un sistema te proporcione una solución que realmente no quiere el usuario.

4.2.2. Abstracción de datos y asociación heurística

En nuestro caso, nuestro sistema depende del input de un usuario, de modo que hay que traducir sus respuestas a datos relevantes para el sistema. Cada elemento de nuestra ontología cuenta con índices que permiten diferenciar entre mejores o peores, o bien con un distintivo en caso de que una respuesta del usuario suponga una limitación.

Por ejemplo, si un usuario dice que quiere un viaje romántico, nuestros elementos de la ontología identificados como románticos tendrán un aumento en su índice de elegibilidad.

Por otro lado, si el usuario dice que quiere hoteles de lujo, se limitarán aquellos alojamientos que se sugieren a sólo aquellos con 4* o más.

4.2.3. Adaptación

A partir de todos los datos obtenidos y el procesamiento, hace falta generar una solución no abstracta. En nuestro caso, cualquier elemento con alta puntuación en el índice de elegibilidad es un buen candidato siempre que cumpla con las restricciones del problema (los días del viaje, el presupuesto, etc.)

4.2.4. Conjunto de operadores

Con este problema contamos con un único operador sobre la solución. Esto es debido a que, debido al diseño heurístico con el índice de elegibilidad, una estancia que cumpla con las limitaciones siempre va a ser una buena candidata, de modo que tenemos una lógica pesada sobre las restricciones para compensar el hecho de tener un único operador.

Operador Añadir Estancia: Añade una estancia generada a partir de una ciudad seleccionada y un subconjunto de sus alojamientos y puntos de interés afines, a demás de el precio invertido en dicho proceso siempre y cuando **TODO** cumpla con las restricciones.

4.2.5. Restricciones de elección de operadores

Al tener un único operador, sus restricciones son todas las restricciones globales relevantes, a demás de que la ciudad de la estancia no haya sido visitada previamente. En el caso del presupuesto, se calcula el presupuesto diario permitido para una ciudad y se comprueba que cumpla con lo requerido.

5. Implementación

Para la implementación de nuestro modelo hemos utilizado la herramienta Protegè para diseñar la ontología, y CLIPS para utilizarla e implementar la lógica.

5.1. Implementación de la ontología

5.2. Módulos

Para solucionar los diferentes subproblemas en CLIPS, hemos decidido dividir la lógica en diferentes módulos.

5.2.1. Módulo MAIN

El módulo principal contiene funciones generales que se utilizan en los otros módulos. Estas funciones son las siguientes:

- **ask-origen:** Función que pregunta la ciudad de origen al usuario y comprueba que la ciudad que responde exista en nuestra base de instancias.
- **ask-question:** Función general que recibe como argumento un string con el texto de la pregunta y realiza una pregunta de si o no al usuario.
- **si-o-no-p:** Función general que recibe como argumento un string con el texto de la pregunta y realiza una pregunta de si o no al usuario.
- **pregunta-numerica:** Función general que recibe como argumentos un string con el texto de la pregunta y un rango (de la forma [minimo valor, maximo valor]) y realiza una pregunta al usuario de respuesta numérica.
- **print-solucion:** Función general que escribe una solución por consola

Este módulo también define todas las variables globales que necesitaremos en el resto de módulos, exporta todo su contenido y contiene una única regla que activa el módulo de preguntas.

5.2.2. Módulo preguntas

Nuestro segundo módulo se encarga de obtener toda la información necesaria del usuario a partir de preguntas. Contiene un conjunto de reglas, cada una de las cuales hace una pregunta y guarda la respuesta en una de las variables globales del MAIN. Estas preguntas son:

- **Origen:** Pregunta al usuario cuál es su ciudad de origen, utiliza la función del MAIN para comprobar que la respuesta tenga sentido y guarda el valor.
- **Continente:** Pregunta al usuario a qué continente le gustaría viajar.
- **Presupuesto:** Pregunta el presupuesto al usuario en un rango de 100 a 20000.
- **Sacrificar calidad:** Pregunta al usuario si permite que se sacrifique calidad de la solución a cambio de ajustar el presupuesto, o no. En caso negativo, asignamos un 20 por ciento extra al presupuesto que ha proporcionado el usuario para que haya más margen y se puedan cumplir mejor las preferencias del usuario y cogeremos siempre el último vuelo de vuelta.
- **Cantidad ciudades:** Pregunta al usuario cuántas ciudades quiere visitar y guarda la información.
- **Días mínimos:** Pregunta al usuario los días que quiere viajar como mínimo y guarda la información.

- **Días máximos:** Pregunta al usuario los días que quiere viajar como máximo. Una vez tiene los días mínimos y máximos, como por diseño hemos decidido que se dividen los días entre las ciudades por igual, buscamos el valor más cercano a la media de días que es divisible entre el número de ciudades, y lo guardamos para utilizarlo en el resto de módulos.
- **Viajar en avión:** Pregunta al usuario si quiere viajar en avión y lo guarda.
- **Viajar en barco:** Pregunta al usuario si quiere viajar en barco y lo guarda.
- **Viajar en tren:** Pregunta al usuario si quiere viajar en tren y lo guarda.
- **Viajar en bus:** Pregunta al usuario si quiere viajar en bus y lo guarda.
- **Viajar de lujo:** Pregunta al usuario si quiere viajar de lujo y lo guarda.
- **Sitios poco conocidos:** Pregunta al usuario si prefiere visitar sitios poco conocidos y lo guarda.
- **Evento especial:** Pregunta al usuario si viaja por un evento especial y lo guarda.
- **Objetivo diversión:** Pregunta al usuario si busca diversión en su viaje y lo guarda.
- **Objetivo cultura:** Pregunta al usuario si busca cultura en su viaje y lo guarda.
- **Objetivo relajación:** Pregunta al usuario si busca relajación en su viaje y lo guarda.
- **Objetivo gastronomía:** Pregunta al usuario si busca gastronomía en su viaje y lo guarda.
- **Objetivo romance:** Pregunta al usuario si busca romance en su viaje y lo guarda.
- **Viaje solo:** Pregunta al usuario si viaja solo y lo guarda.
- **Viaje con amigos:** Pregunta al usuario si viaja con amigos y lo guarda.
- **Viaje con familia:** Pregunta al usuario si viaja con familia y lo guarda.
- **Viaje en pareja:** Pregunta al usuario si viaja en pareja y lo guarda.

Las reglas de las preguntas tienen un orden preestablecido utilizando el saliente de clips, y además hay reglas que no se ejecutan dependiendo de lo que el usuario haya respondido antes. Por ejemplo, si un usuario viaja solo, no se le pregunta si viaja con familia, amigos, etc., como es lógico. La última regla de este módulo activa el siguiente, el módulo de filtrado.

5.2.3. Módulo filtro

La función principal del tercer módulo es eliminar aquellas instancias que no cumplan las restricciones que ha especificado el usuario. Una vez eliminadas, analiza las restantes y aumenta la puntuación de aquellas instancias que mejor cumplan las preferencias de nuestro cliente. Para conseguir esto, se utilizan un conjunto de reglas:

- **Reglas eliminar para eliminar instancias que no cumplan las restricciones de presupuesto:** Tenemos una regla que elimina los transportes que se pasan del presupuesto.
- **Reglas eliminar para eliminar instancias que no cumplan las restricciones de transportes:** Tenemos un conjunto de reglas, una por cada tipo de transporte, que eliminan todas las instancias de los tipos de transporte que no quiera utilizar nuestro usuario.
- **Reglas eliminar para eliminar instancias que no cumplan las restricciones de continente:** Tenemos una regla que marca todas las instancias de ciudad que estén fuera del continente que ha dicho el usuario.
- **Reglas de aumento de puntuaciones:** Por último, una vez eliminadas las ciudades y transportes que no nos interesan, tenemos un conjunto de reglas que, o de forma automática o mediante información de experto, analizan las diferentes instancias y aumentan su puntuación interna para que más adelante, en el módulo de construcción, se puedan ordenar de forma correcta.

Como siempre, tenemos una regla final que activa el siguiente módulo.

5.2.4. Módulo construcción

Nuestro cuarto módulo, el módulo de construcción de la solución, es el más importante ya que contiene toda la lógica que genera el viaje que daremos como resultado al cliente. Se compone de dos partes principales, una primera parte que calcula la media del precio de los transportes y activa unos hechos necesarios después, y la segunda parte que construye la solución ciudad a ciudad. En la primera parte tenemos tres reglas. La primera regla activa dos hechos, que utilizamos para poder reactivar la regla que construye la solución en tiempo de ejecución. Esto sucede cada vez que añadimos una ciudad a la solución para tener en cuenta en todo momento la ciudad de la que se va a salir.

Las otras dos reglas sirven para calcular la media de los transportes, y hacen algunas operaciones con ésta. Primero, obtenemos todos los transportes que cumplen las restricciones del cliente, los contamos y sumamos su precio. Después, en la segunda regla, dividimos el precio total entre el número de viajes que hemos calculado y restamos el resultado a nuestro presupuesto. Una vez tenemos el presupuesto menos la media de los vuelos del continente, dividimos este presupuesto entre nuestro número de ciudades. Hemos decidido hacerlo así para que cada ciudad tenga el mismo presupuesto inicial, y le restamos una media de los transportes para tener margen para volver a casa desde cualquier ciudad, pero ajustando el presupuesto todo lo que podemos. Una vez ejecutadas las reglas de la primera parte, entramos en la segunda parte del módulo de construcción, que tiene una única regla que se va actualizando a partir de un hecho que indica en qué ciudad nos encontramos. Esta regla hace lo siguiente:

- A partir de la ciudad en que nos encontramos, obtiene una lista con todos los transportes hacia otras ciudades que no hayan sido visitadas y que cumplen con las restricciones. Una vez obtenida la lista, la ordena según la puntuación de las ciudades de destino de cada viaje, de mayor a menor.
- Por cada viaje, en orden, comprueba que el vuelo de ida no se pase del presupuesto por ciudad que hemos calculado previamente. En caso de que no se pase, obtiene una lista de los alojamientos de la ciudad de destino y ordena esta lista por calidad del alojamiento de mayor a menor en caso de que sea un viaje de lujo, o por precio de menor a mayor en caso que sea un viaje normal. Como el precio de nuestros hoteles no necesariamente se corresponde con su calidad, hemos decidido ordenar así para que, a no ser que quieras el mayor lujo posible, te encuentre viajes baratos.
- Por cada alojamiento, en orden, comprueba que el precio que costaría quedarnos en ese alojamiento durante los días que hemos calculado previamente no se pase del presupuesto de la ciudad. En caso de que no se pase, obtiene una lista de los puntos de interés de la ciudad y los ordena según la puntuación que hemos calculado, de mayor a menor.
- Por último, por cada punto de interés, y con un máximo de dos puntos de interés por día en la ciudad, comprueba que el precio de visitar el punto de interés no se pase del presupuesto, y en caso que no se pase se lo guarda.
- En caso que sea la última ciudad del viaje, guardamos el transporte de vuelta a la ciudad de origen del cliente, siempre y cuando éste no se pase del presupuesto que habíamos previsto, es decir la media de los transportes del continente.

Si se cumplen todas las condiciones, se genera una instancia de Estancia donde se guardan el transporte de ida, la ciudad, el alojamiento y los puntos de interés. En caso que sea la última ciudad se guarda también el viaje de vuelta. Una vez guardada la Estancia, se actualiza el hecho que marca la ciudad actual, y así la regla se activa de nuevo, hasta que hemos obtenido todas las ciudades que el cliente pedía.

Este módulo se llama tanto para la primera como la segunda solución, así que tenemos unas reglas que reinician algunos valores para que cuando se vuelva a entrar a este módulo por segunda vez se calcule todo correctamente.

5.2.5. Módulo print solución

Este módulo es muy simple, simplemente llama a la función del MAIN print-solucion una vez se ha obtenido la primera solución, y luego activa el módulo que contiene la lógica para calcular la segunda solución correctamente.

5.2.6. Módulo lógica segunda solución

El módulo de lógica de la segunda solución se encarga de minimizar el número de veces que no se encuentra un segundo viaje con las preferencias y restricciones del usuario. Tenemos una regla que comprueba que queden instancias suficientes en el continente objetivo para construir otro viaje con todas las ciudades diferentes, y en caso de que no encontremos suficientes, cambiamos el continente objetivo a otro. Esta decisión se notifica al usuario. Una vez comprobadas estas cosas y cambiado el continente si era necesario, se reinician los valores necesarios y se activa de nuevo el módulo de construcción de solución.

5.2.7. Módulo print segunda solución

Por último, una vez construida la segunda solución, este módulo se encarga de llamar a la función del MAIN para imprimirla por pantalla.

6. Juegos de prueba

6.1. Prueba 1

6.1.1. Descripción del caso

Una familia que vive en Barcelona y quiere hacer un viaje con hijos con un presupuesto moderado de unos 2000 euros. La familia preferiría viajar por Europa a un par de ciudades y como medio de transporte utilizar aviones para que los viajes entre ciudades no se alarguen mucho. Al ir con niños quieren un viaje divertido pero también les gustaría tener un momento romántico para los padres. No viajan por ningún motivo especial. No buscan ningún viaje de lujo pero tampoco les importa si el viaje se va un poco de presupuesto con tal de que el viaje cumpla sus expectativas. Les gustaría viajar alrededor de una semana y por tanto tienen un mínimo de días de 5 y un máximo de 10, ya que sino los niños perderían demasiado colegio.

6.1.2. Solución esperada

Se espera un viaje de mas o menos una semana de duración que sea por Europa. Seguramente con este presupuesto y el hecho de que el viaje se vaya a desarrollar en el continente de partida, el precio de los vuelos sera menor, y sin tener requerimientos de viaje de lujo los hoteles estarán mas o menos en 2 estrellas. Los puntos de interés a visitar pueden ser parques de atracciones o otras experiencias divertidas combinadas con lugares románticos.

6.1.3. Resultado obtenido

Solución 1:

Estancia en: Paris, 3 dias

Transporte: Avión

Alojamiento: Mimosa 1*

Puntos de interes: Disneyland, Museo del Louvre, Torre Eiffel, Notre Dame

Estancia en: Dinant, 3 dias

Transporte: Avión

Alojamiento: La Fayette 2*

Puntos de interes: Castillo de Veves, Dinant Citadel, Parque Furfooz

Transporte de vuelta: Avión

Precio final: 970.0

Numero de días: 6

Solución 2:

Estancia en: Berlin, 3 dias

Transporte: Avión

Alojamiento: Gotland 2*

Puntos de interes: Holocaust-Mahnmal, Puerta de Brandenburg y Reichstag Building

Estancia en: Bergen, 3 dias

Transporte: Avión

Alojamiento: Edvard Grieg 4*

Puntos de interes: Ulriken, Bryggen, Floyen

Transporte de vuelta: Avión

Precio final: 1217.0

Numero de días: 6

Como podemos ver, la primera solución es perfecta para esta familia y se ajusta a lo esperado. La segunda opción, aunque no es completamente mala, al haberse hecho con los restos de instancias difiere un poco del objetivo del viaje aunque los requisitos de presupuesto, la zona donde viajar, y el transporte a utilizar siguen siendo los deseados.

6.2. Prueba 2

6.2.1. Descripción del caso

Una pareja de gente rica , originarios de Nueva York, quiere un viaje completo sin preocuparse por ningún gasto. Su presupuesto es del máximo permitido por la agencia, en este caso 20000, y no les importa si se sobrepasa. El motivo del viaje es la reciente boda de la pareja y les gustaría ir a disfrutar de los sitios mas románticos, relajantes y las mas grandes experiencias que pueda ofrecer Asia. Solo quieren viajar en avión ya que cualquier otra cosa les haría parecer gente de baja clase y por la misma razón quieren que su viaje sea lo mas lujoso posible. El viaje tiene que ser del máximo de días posible y visitar el máximo de ciudades ya que quieren empaparse de las diferentes culturas del continente.

6.2.2. Solución esperada

La solución tiene que ser lo mas lujosa posible, visitar el mayor numero de ciudades y gastar el máximo del presupuesto posible. Las ciudades y puntos de interés a visitar serán balnearios, spas y otros sitios que cumplan los requisitos de románticos y relajantes.

6.2.3. Resultado obtenido

Solución 1:

Estancia en: Tokio, 6 dias

Transporte: Avión

Alojamiento: Roppongi Tokyo Premier 5*

Puntos de interés: Aguas termales de Tokio, Akihabara, Mercado de pescado de Toyosu, Shibuya, Templo Sensoji y Torre de Tokio

Estancia en: Bangkok, 6 dias

Transporte: Avión

Alojamiento: Eastin Grand 5*

Puntos de interés: Barrios Rojos, Canales de Bangkok, Wat Arun, Rooftop bars, Chinatown, Wat Pho y Khao San

Estancia en: Incheon, 6 dias

Transporte: Avión

Alojamiento: Gran Hyatt 5*

Puntos de interés: Aguas termales Ganghwado, Songdo Central Park, Songwol don Fairytale Village, Jeondeungsa Temple, Sinpo Internation Market

Estancia en: Katmandu, 6 días

Transporte: Avión

Alojamiento: Apsara 4*

Puntos de interés: Barrio de Thamel, Durbar Square, Parque de los tres budas, Estupa de Boudhanath

Transporte de vuelta: Avión

Precio final: 4514.0

Numero de días: 24

Solución 2:

Estancia en: Paris, 6 días

Transporte: Avión

Alojamiento: National Des Arts et Metiers 5*

Puntos de interés: Torre Eiffel, DisneyLand, Museo Louvre, Notre Dame

Estancia en: Dinant, 6 días

Transporte: Avión

Alojamiento: Les Jardins de la Molinee 4*

Puntos de interés: Parque Furfooz, Dinant Citadel, Castillo Veves

Estancia en: Barcelona, 6 días

Transporte: Avión

Alojamiento: W 5*

Puntos de interés: Sagrada Familia, Parc Guell, Mercat Boqueria

Estancia en: Berlin, 6 días

Transporte: Avión

Alojamiento: TITANIC Chausse 4*

Puntos de interés: Holocaust Mahmmal, Puerta de Brandenburg, Reichstag Building

Transporte de vuelta: Avión

Precio final: 11667.0

Numero de días: 24

Como podemos observar en la solución 1 el viaje esta compuesto por lo mas lujoso y caro posible. Al tener tanto dinero para gastar se les permite hacer un montón de actividades y al tener un gran numero de días es posible disfrutar de mas puntos de interés hasta de los que cumplen los requisitos. La solución 2 también cumple casi todos los requisitos exceptuando el continente, estos es debido a que al no tener suficientes instancias se proporciona una solución en otro continente.

6.3. Prueba 3

6.3.1. Descripción del caso

Un grupo de amigos acaba de terminar el bachillerato y quiere ir de viaje de fin de curso. Han estado ahorrando un par de años con el fin de este viaje y tienen 3000 euros de presupuesto, aún así son conscientes de que queriendo visitar 4 ciudades no pueden pedir el gran lujo. Están dispuestos a utilizar medios de transporte mas baratos y mas largos con el fin de poder ajustarse al presupuesto y por tanto no quieren

viajar en avión. El viaje esta principalmente a evadirse de los estudios y por tanto buscan el máximo de diversión, experiencias posible. También están interesados por la gastronomía, la cultura y ver rincones inexplorados que sean menos conocidos para el publico. Son originarios de Dinant y su idea es que el viaje se sitúe por el continente Europeo, para ello quieren estar entre 15 y 25 días y visitar 4 ciudades distintas.

6.3.2. Solución esperada

Se espera que los alojamientos provistos por la solución sean de la mas baja calidad y con el menor precio posible con tal de poder realizar el viaje. También se espera que se viaje mas con transportes como el bus y el tren ya que son mas económicos y el grupo de amigos ha preferido no gastar tanto del presupuesto en vuelos. También se espera que en la solución aparezca resultados poco conocidos.

6.3.3. Resultado obtenido

Solución 1:

Estancia en: Barcelona, 5 dias

Transporte: Bus

Alojamiento: Generator-Barcelona 1*

Puntos de interes: Mercat Boqueria, Parc Güell, Sagrada Familia

Estancia en: Berlin, 5 dias

Transporte: Bus

Alojamiento: Gotland 2*

Puntos de interes: Holocaust-Mahnmal, Puerta de Brandenburg, Reichstag Building

Estancia en: Paris, 5 dias

Transporte: Bus

Alojamiento: Alojamiento Mimosa 1*

Puntos de interes: Disneyland, Museo Louvre, Torre Eiffel, Notre Dame

Estancia en: Bergen, 5 dias

Transporte: Bus

Alojamiento: Edvard Grieg 4*

Puntos de interes: Ulriken, Bryggen, Floyen

Transporte de vuelta: Bus

Precio final: 2549.0

Numero de días: 20

Solución 2:

Estancia en: —

Transporte: —

Alojamiento: —

Puntos de interes: —

Transporte de vuelta: —

Precio final: —

Numero de días: —

Esta vez solo conseguimos una opción de viaje ya que con el presupuesto acordado, el requerimiento de no utilizar aviones y el hecho de que tendríamos que buscar la solución dentro de otro continente ya que no quedarían suficientes en este para dar otra, nos vemos en la situación de que no hay combinación posible para esto. En el caso de la solución que si nos devuelve el SBC podemos observar que los hoteles corresponden a lo esperado, bajo nivel, exceptuando Bergen donde el hecho de coger un hotel de 4 estrellas es debido a que

no hay instancias de menor nivel y que al haber sobrado presupuesto de las otras ciudades les permite a este grupo de amigos darse un pequeño lujo al final del viaje.

6.4. Prueba 4

6.4.1. Descripción del caso

Un hombre de Jaipur quiere hacer un viaje solo. Tiene un presupuesto de unos 400 euros, quiere viajar entre 3 y 5 días y quiere ir a 1 sola ciudad. Tiene miedo a las alturas así que no quiere pillar avión. Le gustaría visitar su propio continente, Asia, para descubrir mas sobre la cultura y los sitios menos conocidos. También le interesa la cultura gastronómica.

6.4.2. Solución esperada

La solución esperada es que visite una ciudad poco conocida y descubra los puntos de interés que le pueda ofrecer esta. El hospedaje tendría que ser de una calidad media tirando a baja con tal de poder satisfacerse con ese presupuesto y los días de viaje serán unos 4. Es probable que no se pueda dar una segunda solución ya que el presupuesto es un poco ajustado.

6.4.3. Resultado obtenido

Solución 1:

Estancia en: Katmandu, 4 días

Transporte: Tren

Alojamiento: Sweet Dreams 0*

Puntos de interés: Durbar Square, Parque de los Tres Budas, Estupa de Bpudhanath, Barrio de Thamel

Transporte de vuelta: Tren

Precio final: 234.0

Numero de días: 4

Solución 2:

Estancia en: —

Transporte: —

Alojamiento: —

Puntos de interés: —

Transporte de vuelta: —

Precio final: 0.0

Numero de días: —

Como se podía esperar en la primera solución hemos encontrado un viaje a medida para nuestro cliente donde se hospeda en un alojamiento muy barato, ya que el presupuesto era muy escueto, y visita una buena cantidad de puntos de interés culturales. Al ser un presupuesto tan limitado nos encontramos que no hay otra ciudad que cumpla con los requisitos y por tanto se pinte un error de que no ha sido posible.

6.5. Prueba 5

6.5.1. Descripción del caso

Una mujer empresaria procedente de Tokio quiere darse un buen respiro del trabajo y dar un viaje alrededor de Asia. Le gustaría visitar 4 ciudades en un periodo entre 10 y 20 días y además quiere un viaje de lujo. Esta especialmente interesada en la gastronomía, la cultura y el descanso. Le da igual si visitar lugares desconocidos y tiene un presupuesto de unos 8000 euros.

6.5.2. Solución esperada

Se espera un viaje con hoteles caros y con el máximo de estrellas posible, unos puntos de interés relacionados con los tópicos como pueden ser bares, estatuas y templos.

6.5.3. Resultado obtenido

Solución 1:

Estancia en: Bangkok, 3 dias

Transporte: Avion

Alojamiento: Eastin Grand 5*

Puntos de interés: Chinatown, Wat Arun, Wat Pho, Barrios Rojos, Khao San, Rooftop bars

Estancia en: Katmandu, 3 dias

Transporte: Avion

Alojamiento: Apsara 4*

Puntos de interés: Durbar Square, Parque de los Tres Budas, Estupa de Bpudhanath, Barrio de Themel

Estancia en: Incheon, 3 dias

Transporte: Avion

Alojamiento: Grand Hyatt 5*

Puntos de interés: Aguas termales Ganghwado, Jeondeungsa Temple, Songdo Central Park, Sinpo International Market, Songwol don Fairytale Village

Transporte de vuelta: Avion

Precio final: 2076.0

Numero de días: 9

Solución 2:

Estancia en: Bergen, 3 dias

Transporte: Avion

Alojamiento: Opus 16 5*

Puntos de interes: Ulriken, Bryggen,Floyen

Estancia en: Barcelona, 3 dias

Transporte: Avion

Alojamiento: W 5*

Puntos de interes: Mercat Boqueria, Parc Güell, Sagrada Familia

Estancia en: Paris, 3 dias

Transporte: Avion

Alojamiento: Lutetia 5*

Puntos de interes: Museo Louvre, Torre Eiffel, Notre Dame, Disneyland

Estancia en: Berlin, 4 dias

Transporte: Avion

Alojamiento: Adlon Kempinski 5*

Puntos de interes: Holocaust-Mahnmal, Puerta de Brandenburg, Reichstag Building

Transporte de vuelta: Avion

Precio final: 8229.0

Numero de días: 12

En este caso la solución 1 tiene el problema de que al pedir un viaje de lujo como Asia y con el máximo de ciudades partiendo de una ciudad que no es Jaipur no se puede satisfacer los requerimientos y te devuelve una solución incompleta. Esto es debido a que las instancias de hoteles de Jaipur no tienen hoteles de lujo (ya que en la ciudad como tal no abundan) y aunque el presupuesto te lo permita el filtro borra como opción la propia ciudad. La solución 2 es acorde con lo esperado.

7. Conclusiones

Esta práctica ha sido nuestro primer intento de hacer un sistema de experto basado en el conocimiento. Como tal, hemos tomado decisiones de cara al sistema que quizás no sean las más adecuadas para generar una solución, pero todas estas decisiones han sido tomadas desde el criterio de experto”de nosotros los desarrolladores, ya que no contábamos con un experto real. Dicho esto, nos ha servido para poder darnos cuenta de la importancia de la existencia de dicho experto después de tener serias dudas en muchas de las decisiones.

Hemos tenido especiales dificultades principalmente con el funcionamiento de CLIPS, dada la falta de familiaridad con su estructura de funcionamiento, y hemos aprendido a usarlo mejor hacia la etapa final del desarrollo, donde hemos podido diseñar de manera más pulida. Este hecho se nota especialmente en las partes iniciales del proyecto ya que empezamos enfocándolo de una manera que dejaba bastante que desear y comparada con los últimos retoques, en los que teníamos un control mucho más preciso de CLIPS, se nota una gran diferencia.

Como hemos comentado antes, interpretamos algunos datos de modo un poco restrictivo, y creemos que puede dar lugar a más situaciones sin solución pero creemos que es un mal necesario para no sugerir soluciones que nosotros consideraríamos incorrectas.

Teniendo todo esto en cuenta creemos que esta primera toma de contacto con un SBC nos ha servido para aprender y realizar uno con una utilidad clara y práctica y estamos claramente satisfechos con el trabajo final.