

# Inteligencia Artificial

---

## Sistemas Basados en el Conocimiento

Jia Long Ji Qiu

Shuang Long Ji Qiu

You Wu

Curso 2021/2022 Q1

# Índice

|  |    |
|--|----|
| 1. Identificación                          | 3  |
| 1.1. Descripción del problema              | 3  |
| 1.2 Viabilidad de construir SBC            | 4  |
| 1.3. Fuentes de conocimiento               | 4  |
| 1.4. Objetivos y resultados esperados      | 5  |
| 2. Conceptualización                       | 6  |
| 2.1. Conceptualización del problema        | 6  |
| 2.2. Elementos y conceptos del dominio     | 6  |
| 2.3. División de subproblemas              | 8  |
| 2.4. Ejemplos del conocimiento del experto | 9  |
| 2.5. Proceso de resolución                 | 10 |
| 3. Formalización                           | 11 |
| 3.1. Desarrollo de la ontología            | 11 |
| 3.2. Metodología de resolución             | 16 |
| 4. Implementación                          | 18 |
| 4.1. Construcción de la ontología          | 18 |
| 4.2. Módulos                               | 18 |
| 4.3. Prototipos del problema               | 20 |
| 5. Juegos de prueba                        | 21 |
| 6. Conclusiones                            | 24 |
| 7. Planificación                           | 25 |

# 1. Identificación

## 1.1. Descripción del problema

El objetivo de esta práctica consiste en resolver un problema acerca de viviendas adecuadas mediante el Sistema Basados en el Conocimiento.

El problema que nos plantea es el siguiente: un organismo dispone gran cantidad de ofertas de alquileres, en la cual cada oferta tiene las características más relevantes de la vivienda. A partir de las características del cliente solicitador el sistema genera un listado de las ofertas de alquileres, ordenado de manera descendente por el grado de recomendación.

El enunciado también nos ofrece un listado de servicios que también podría ser un requisito para la solicitud de un alquiler, como por ejemplo que un cliente quiere una vivienda que esté cerca de un servicio de transporte público. Para ello nosotros hemos definido la cercanía según si están cerca (menos de 1000 km) o lejano (más de 1000 km).

Para sus recomendaciones, el sistema se basará en las ofertas de viviendas y las preferencias y los datos del solicitante, de manera que cada uno de ellos recibirá aquellas ofertas que le pueda interesar.

Las principales características del solicitante que tendremos en cuenta son:

- **Datos del solicitante:** datos básicos del solicitador, como la edad del solicitante, tipología de los solicitantes(casado/soltero/pareja/divorciado/con hijos), si está cargo de personas mayores, si tiene coche, si trabaja/estudia en la ciudad(en el caso que sí, indicar la ubicación del servicio).
- **Preferencias y/o restricciones del solicitante:** se trata de identificar las preferencias y/o restricciones del solicitante sobre las viviendas, para poder ajustar de manera más adecuada un conjunto específico de ofertas interesante al cliente.

Una vez obtenidas esas informaciones, el sistema podrá recomendar ofertas de viviendas a aquellos solicitantes que estén interesados.

## 1.2 Viabilidad de construir SBC

El problema consiste en encontrar un listado de ofertas de alquileres de tal manera que ajuste las preferencias y restricciones del cliente. Para crear el listado será necesario hacer un recorrido sobre todo el espacio de las ofertas de alquiler de viviendas del organismo, por lo tanto, podemos decir que es un problema de búsqueda.

La complejidad de este problema, en el cual la información de cada estado es cambiante y una parte de esta puede ser más relevante que otra, hace que no sea viable aplicar un algoritmo genérico, el cual siempre espera la misma información en cada estado y la evalúa de la misma manera. Sin embargo, como tenemos un conocimiento previo sobre el contexto de cada estado (datos del solicitante, preferencias y/o restricciones del solicitante), podemos expresar mediante el Sistema Basados en Conocimiento ciertas reglas que nos permitirán hacer un recorrido más efectivo sobre el espacio de las ofertas de alquileres, pues podremos tomar mejores decisiones sabiendo el grado de relevancia de toda esta información.

Podemos concluir, pues, que utilizando un SBC para resolver este problema, en el que siempre tendremos un conocimiento dependiente del dominio, nos permitirá obtener una solución para cada cliente en específico de manera eficiente.

## 1.3. Fuentes de conocimiento

Las fuentes del conocimiento son el pináculo de un Sistema Basado en el Conocimiento ya que es por medio de esta el sistema aprende y es capaz de tomar decisiones. Para construir nuestro sistema tenemos las dos siguientes fuentes principales:

- **Los clientes:** es la fuente de conocimiento que obtendremos mediante la ejecución del programa y será importante para poder generar el listado de ofertas, cuanto más información tengamos sobre ellos mejor y por lo tanto el listado de recomendaciones se ajustará más en el gusto de los clientes. La obtención de esta información será por medio de un conjunto de preguntas y el usuario indicará cuáles de estas serán restrictivas o preferentes.
- **Las características de los alquileres:** también es un factor muy importante, ya que a partir de ellos obtendremos la información necesaria para poder relacionar los alquileres con las preferencias y restricciones de los clientes. Esta información estará almacenada en la ontología en la cual habremos introducido todos los datos de los alquileres disponibles en ese momento.

Aparte de las dos fuentes mencionadas anteriormente, también utilizaremos el conocimiento propio.

## **1.4. Objetivos y resultados esperados**

Los objetivos principales serán:

- El sistema debe interactuar con el cliente para obtener la información necesaria respecto a los datos del cliente junto con sus restricciones y las preferencias sobre la vivienda.
- Descartar aquellas viviendas en las que no cumple el mínimo número posible de restricciones que haya indicado el cliente (en el caso que haya).
- Ordenar aquellas viviendas que no han sido descartadas mediante las preferencias del usuario para más tarde clasificarlas según el grado de recomendación.
- Finalmente, el sistema deberá imprimir la recomendación.

El resultado que se espera obtener será una lista de viviendas recomendadas para el cliente que, como ya hemos comentado anteriormente, saldrá según el grado de recomendación.

Para cada vivienda se detallan las razones por la cual se recomienda dicha vivienda.

Respecto al grado de recomendación, serán los siguientes:

- Parcialmente adecuado: no se cumplen completamente uno o dos criterios pero podría ser aceptable.
- Adecuado: cumple todos los requerimientos.
- Muy recomendable: cumple todos los requerimientos y además tiene características extra que lo hacen más ventajoso.

## 2. Conceptualización

### 2.1. Conceptualización del problema

En la conceptualización del problema se pretende obtener una visión del problema desde el punto de vista del experto y por lo tanto permitimos obtener el conocimiento adecuado para utilizarlo y así resolver el problema que se plantea.

Por lo tanto, la realización de esta tarea se dividirá en 4 fases:

- **Primera fase.** Debemos conocer los elementos, conceptos que debe de tratar nuestro sistema y sus características para así en la siguiente fase poder construir una ontología que los formalice.
- **Segunda fase.** Se pretende obtener una descomposición del problema principal es subproblemas, realizando en cada una de ellas un análisis de refinamiento sucesivos hasta que nos podamos hacer una idea de la relación jerárquica de las diferentes fases de resolución hasta los operadores de razonamiento más elementales.
- **Tercera fase.** Daremos algunos ejemplos del conocimiento del experto que será extraído de la información que proporciona el enunciado del problema y de esta manera podremos tener una idea aproximada de cómo resolver el problema desde el punto de vista del experto.
- **Cuarta fase.** Las fases anteriores, si lo hemos hecho bien, nos debería de dar un modelo casi formal del dominio, de los problemas y métodos de resolución que se tendría que incluir en nuestro sistema.

### 2.2. Elementos y conceptos del dominio

En nuestro sistema/base de datos del proyecto contendrá principalmente diferentes instancias de viviendas, dormitorios y diferentes servicios, donde cada una de estas tendrá ciertos atributos que describen sus propiedades y así poder hacer un filtrado. Estos atributos los detallaremos más adelante.

En el sistema debe tener la siguiente información:

### **1. Características de una vivienda:**

Para poder describir todas las propiedades que nos permiten describir adecuadamente una vivienda, necesitamos conocer cada uno de las siguientes características:

- El precio mensual del alquiler (€/mes).
- Localización de la vivienda.
- Superficie habitable de la vivienda.
- Número de dormitorios que dispone y además si son simples o dobles.
- Algunas características adicionales de las viviendas como por ejemplo si tiene balcón, terraza, garaje o si tiene instalado aire acondicionado.
- El tipo de la vivienda: unifamiliar, dúplex o piso.
- Detalles extras como si hay normativas determinadas como es si se puede tener mascotas en la vivienda.

### **2. Características de un servicio:**

Para poder describir todas las propiedades que nos permiten describir adecuadamente un servicio necesitamos conocer cada uno de las siguientes características:

- Localización del servicio
- El tipo del servicio, por ejemplo, si es un servicio de ocio, un servicio de comercio o si es un servicio educativo.

### **3. Características de una oferta:**

Para poder realizar una búsqueda que atienda efectivamente a las demandas de viviendas, se requiere conocer las siguientes características sobre la petición:

- Precio mínimo y máximo que está dispuesto a ofrecer el cliente por un alquiler.
- El número de dormitorios que tiene la vivienda.
- El tamaño de los dormitorios.
- Cercanía o lejanía a ciertos servicios que se encuentren alrededor de la vivienda.
- Edad de los usuarios.
- Estado civil en el que se encuentra el cliente.
- Lugar de trabajo o estudio en caso de que trabaje o estudie.

Cabe mencionar que las características de las ofertas, representan mayoritariamente las restricciones o preferencias de las peticiones que las hemos definido nosotros.

A parte de los conceptos mencionados anteriormente, cabe destacar las recomendaciones, otro término importante para la resolución del problema. Las recomendaciones serán básicamente un medio para almacenar las soluciones que hayamos computado.

## **2.3. División de subproblemas**

Para la resolución del objetivo principal, hemos dividido el procedimiento en diferentes subproblemas, concretamente tenemos 4 subproblemas que se ejecutarán de forma secuencial.

### **1. Recopilación de datos**

La primera tarea que se lleva a cabo es realizar un conjunto de preguntas al cliente. Estas preguntas que se hacen serán las necesarias para poder elaborar un conocimiento preciso de su solicitud, sus restricciones y sus preferencias. Tanto los datos de las viviendas como los de los servicios estarán en la base de datos, o sea, serán instancias del sistema.

### **2. Procesamiento de datos**

Una vez hayamos obtenido los datos necesarios en la fase de recopilación y almacenado en el sistema, necesitamos analizarlos y abstraer conocimiento nuevo para llegar al objetivo.

Esta fase será de vital importancia ya que es aquí donde, a parte de analizar la información, también determinaremos qué viviendas seleccionaremos como solución del problema. Para ello primero obtenemos todas las instancias de viviendas que tengamos y los filtramos con las restricciones. Para toda vivienda del sistema, en el caso de que no cumpla el mínimo número de restricciones admisible será descartada como parte de la solución.

Después, las viviendas que hayan sobrevivido al filtrado serán las recomendaciones que el SBC pueda ofrecer y se pasará a evaluar que tan buena es la solución mediante las preferencias que haya indicado el cliente.

Se valorará de forma positiva aquellas viviendas que cumplan las preferencias aumentando la puntuación de la vivienda y, en el caso contrario, simplemente no lo aumentamos pero nunca descartamos.



### **3. Presentación de la solución**

Finalizada el proceso de filtrado, se mostrará la solución según el grado de recomendación (muy recomendable, adecuado, parcialmente adecuado) de manera ordenada según la puntuación que hemos calculado para cada vivienda.

Por una parte, en el caso de que el grado de recomendación sea parcialmente adecuado, también indicaremos qué criterios no se ha podido cumplir.

Por otra parte, en el caso de que el grado de recomendación sea muy recomendable, también indicaremos qué características hace que destaque sobre las otras.

### **2.4. Ejemplos del conocimiento del experto**

Para realizar gran parte de la conceptualización, es necesario ver cómo se resuelven ciertos criterios desde el punto de vista del experto. A partir del enunciado, podemos abstraer hechos generales que podrán ser aplicados dependiendo del contexto.

Algunos ejemplos podrían ser los siguientes:

- Si el cliente está a cargo de personas mayores, entonces podemos deducir que interesará viviendas que están, por ejemplo, cercanas a servicios de sanidad.
- Del ejemplo anterior, podemos extraer también que si el cliente es mayor entonces les puede importar estar cerca de servicios sanitarios.
- Si el cliente es joven, entonces podemos deducir que le puede interesar estar cerca de servicios de ocio.
- Si el cliente no tiene coche, podemos recomendar viviendas que estén cerca de estaciones de transporte público.
- Si el cliente tiene hijos, podemos recomendar viviendas que estén cerca de colegios, institutos o universidades, es decir servicios de educación.
- Si el cliente tiene pareja o conforma una familia, entonces se puede deducir que le interese estar cerca de servicios comerciales.
- Si el cliente quiere un entorno tranquilo, podemos deducir que no quiere estar cerca de zonas industriales.

## **2.5. Proceso de resolución**

En el proceso de resolución detallaremos de manera informal el flujo de razonamiento, este será el siguiente:

El cliente responderá un conjunto de preguntas del sistema, indicando de manera explícita si la respuesta que da es restrictiva (obligatoriedad positiva o obligatoriedad negativa), preferente o si le es indiferente sobre las viviendas.

Además de estas preguntas, se le preguntarán algunos aspectos personales para que el sistema tenga un perfil aproximado del cliente.

Una vez que el cliente haya introducido algunos datos personales, sus restricciones, sus preferencias y sus indiferencias, pasaremos a la fase de abstraer las viviendas que pueden ser recomendadas.

Una vez completada la abstracción de las viviendas se descartará todas aquellas viviendas que incumplan más de dos restricciones impuestas por el cliente, tal como indica el enunciado. Nótese que las viviendas están instanciadas en el sistema.

Después de obtener las viviendas, pasamos a la fase de puntuar estas viviendas por su calidad, tendremos por lo tanto las que son muy recomendadas, las adecuadas y las que son parcialmente adecuadas. Además debemos imponer un orden dentro de cada grado de recomendación y finalmente, ilustrar el resultado.

### 3. Formalización

#### 3.1. Desarrollo de la ontología

Para el desarrollo de la ontología, seguiremos las fases que hemos visto en la clase de teoría de la asignatura.

##### **Fase 1. Determinar el dominio y la cobertura de la ontología.**

En esta fase debemos plantearnos algunas preguntas básicas para poder realizar una ontología consistente.

Las preguntas serán las siguientes:

- ¿Qué dominio cubrirá la ontología?
- ¿Para qué usaremos la ontología?
- ¿A qué tipos de preguntas ha de poder responder la ontología?
- ¿Quién usará y mantendrá la ontología?

Respecto a la primera pregunta, en el apartado de Conceptualización hemos comentado los conceptos del dominio. Estos mismos conceptos son los que hemos utilizado para crear la ontología. Principalmente los conceptos que nos interesa definir son las viviendas, los servicios, las peticiones y los clientes junto con sus respectivas características.

Respecto a la segunda pregunta, usaremos la ontología para representar las viviendas que se pueden alquilar en el mercado inmobiliario de Barcelona y los clientes que quieran alquilar. Por lo tanto, el primer uso que le daremos a la ontología será un sistema de recomendación de alquileres de viviendas.

Respecto a la tercera pregunta, la ontología deberá responder principalmente a preguntas que estén relacionadas con las viviendas, es decir sus características y también algunas preguntas sobre el propio cliente. Las principales preguntas serán:

- ¿Qué edad tiene?
- ¿Cuál es su estado civil?
- Está a cargo de personas mayores?
- Tiene coche?
- ¿Trabaja en la ciudad de Barcelona?
- En qué distrito trabaja?

- ¿Qué tipo de vivienda está buscando?
- Cuál es el precio mensual mínimo que está dispuesto a pagar?
- Cuál es el precio mensual máximo que está dispuesto a pagar?
- Cuantos metros cuadrados quiere que tenga aproximadamente como mínimo?
- Cuántos dormitorios simples quiere como mínimo?
- De estos dormitorios, ¿cuántos quiere con armarios empotrados como mínimo?
- ¿Cuántos dormitorios dobles quiere como mínimo?
- De estos dormitorios, ¿cuántos quiere con armarios empotrados como mínimo?
- ¿Cuántos baños quiere como mínimo?
- De estos baños, ¿cuántos quiere con bañera como mínimo?
- De estos baños, ¿cuántos quiere con ducha como mínimo?
- Quiere una vivienda con balcones?
- Quiere una vivienda con terrazas?
- Quiere una vivienda unifamiliar con garaje?
- Quiere una vivienda unifamiliar con sótano?
- Quiere una vivienda con ascensor?
- Quiere una vivienda donde se permiten mascotas?
- Quiere una vivienda con gas natural o butano?
- Quiere una vivienda que disponga de aire acondicionado?
- Quiere una vivienda que tenga calefacción?
- Quiere una vivienda amueblada?
- Quiere una vivienda con electrodomésticos?
- Quiere una vivienda con piscina?
- Quiere una vivienda soleada por la mañana, por la tarde, o todo el día?
- Quiere una vivienda con buenas vistas?
- Qué servicios prefiere tener cerca?
- Qué servicios prefiere tener lejos?

Finalmente, respecto a la última pregunta, esta ontología la mantendremos nosotros, ya que la usaremos para desarrollar la práctica de la asignatura. A pesar de todo, cualquier persona podría beneficiarse de este sistema si se requiere de este.

## **Fase 2. Considerar la reutilización de ontologías existentes.**

Para la realización de esta práctica, no hemos considerado utilizar ontologías existentes y únicamente utilizar una que la hayamos creado nosotros mismos en función de los requisitos de la práctica teniendo en cuenta las fuentes de información dadas en el enunciado de la práctica y de las peticiones que queremos realizar.

### Fase 3. Enumerar los términos importantes en la ontología

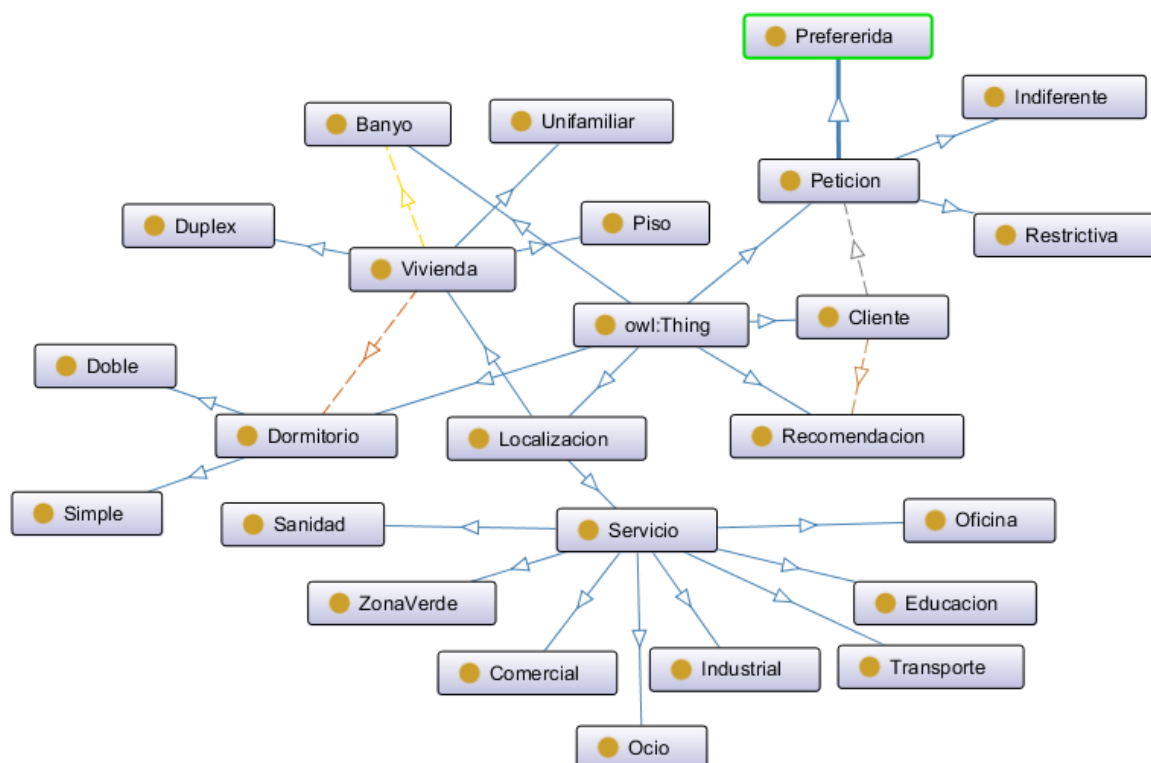
Como ya hemos comentado anteriormente, para poder realizar la ontología debemos identificar ciertos conceptos importantes del dominio, en nuestro caso, necesitaremos saber sobre las características de las viviendas, también necesitamos qué cosas o qué criterios quiere el cliente respecto a esas viviendas. Por último, debemos tener en cuenta la localización de los servicios. Podemos destacar que los conceptos más importantes són las características de las viviendas y las peticiones del cliente.

### Fase 4. Definir las clases y su jerarquía

Como ya hemos dicho anteriormente, los conceptos importantes tales como la vivienda, los servicios, las peticiones y los clientes han de poder representarse.

Primero veremos una visión completa de la ontología y seguidamente iremos viendo cada uno de los conceptos que la forman parte e iremos explicando sus características.

La visión completa de la ontología para esta práctica es la siguiente:



## Localización

Clase que representa un lugar. Sus características son:

- dirección: String, indica la dirección que se encuentra la localización.
- distrito\_ciudad: symbol, indica en qué distrito se encuentra la localización.
- latitud\_y: integer, representa la coordenada y.
- longitud\_x: integer, representa la coordenada x.

Los distritos válidos serán los que haya en Barcelona, lugar donde simularemos la práctica, por lo tanto podrá ser lo siguientes distritos: Ciutat Vella, Eixample, Sants-Montjuïc, Les Corts, Sarrià-Sant Gervasi, Gràcia, Horta - Guinardó, Nou Barris, Sant Andreu y Sant Martí.

Además esta clase tendrá dos subclases que son vivienda y servicio.

## Vivienda

Como ya indica, representa una vivienda. A parte de las características que hereda de localización, este concepto se caracteriza por:

- tiene\_dormitorio: Instancia de Dormitorio
- tiene\_baños: Instancia de Baño
- has\_aire: boolean, true si tiene aire acondicionado, false en otro caso.
- has\_balcon: boolean, true si tiene balcón, false en otro caso.
- has calefaccion: boolean, true si tiene calefacción, false en otro caso.
- has\_terraza: boolean, true si tiene terraza, false en otro caso.
- m2\_habitable: integer, metros cuadrados habitable de la vivienda.
- electrodomésticos: boolean, true si tiene electrodomésticos, false en otro caso.
- gas: symbol, indica si el sistema de gas es con butano o natural.
- has\_ascensor: boolean, true si tiene ascensor, false en otro caso.
- has\_piscina: boolean, indica si tiene piscina (true/false).
- has\_vistas: boolean, indica si tiene vistas (true/false).
- mascotas: boolean, indica si se puede tener mascotas en la vivienda (true/false).
- precio\_mensual: integer, indica el precio mensual que cuesta la vivienda.

Además, dentro de las viviendas distinguimos tres tipos: unifamiliares, pisos y dúplex.

## Unifamiliares

Subclase de Vivienda y representan viviendas unifamiliares. Sus atributos son:

- has\_garaje: boolean, indica si tiene o no garaje.
- has\_sotano: boolean, indica si tiene o no sótano.

## Pisos y Dúplex

Subclase de Vivienda y representan pisos y dúplex. Sus atributos son:

- `has_parking`: boolean, indica si tiene o no parking.
- `altura`: integer, indica la altura de la vivienda (entresuelo, primero, segundo, etc.)

## Servicio

Subclase de Localización, únicamente tendrá los atributos heredados de la superclase: además tiene 8 subclases, que serán los diferentes servicios, estos son: Sanidad, Zona Verde, Comercial, Ocio, Industrial, Transporte, Educación y Oficina. Cabe destacar que estas subclases tampoco tienen atributos propios, para esta práctica no hemos refinado los tipos de servicios, sin embargo, explicaremos los significados de estas subclases.

- **Sanidad**: cualquier tipo de servicio del entorno sanitario, por ejemplo: hospitales, dentistas, centro médico, etc.
- **Zona Verde**: incluye zonas ecológicas tales como parques naturales o bosques.
- **Comercial**: cualquier servicio del entorno comercial, por ejemplo: supermercados, hipermercados, centros comerciales, tiendas, etc.
- **Ocio**: aquí incluiríamos las discotecas, los bares, el cine, etc.
- **Industrial**: cualquier servicio del sector industrial: fábricas, talleres, etc.
- **Transporte**: cualquier servicio de transporte público, como las paradas de autobús.
- **Educación**: incluimos colegios, parvularios, universidades, etc.
- **Oficina**: instituciones, empresas, etc.

## Dormitorio

Clase que representa un dormitorio, este puede ser un dormitorio simple o uno doble que serán sus subclases. Sus características son:

- `armario_empotrado`: boolean, indica si en el dormitorio hay armarios empotrados (true/false).

Las subclases de dormitorio, Dormitorio Simple y Dormitorio Doble no tienen atributos propios, solo heredan de su clase padre.

## Baño

Clase que representa un baño, sus características son:

- `has_baño`: boolean, indica si tiene bañera (true/false).
- `has_ducha`: boolean, indica si tiene ducha (true/false).

## **Peticion**

Clase que representa la petición que hace el cliente. Además de las características de la vivienda, también contiene:

- max\_precio: integer, precio máximo que está dispuesto a pagar el cliente por la vivienda.
- min\_precio: integer, precio mínimo que está dispuesto a pagar el cliente por la vivienda.
- num\_banyos: integer, número de baños que quiere el cliente.
- num\_dormS: integer, número de dormitorios simples.
- num\_dormD: integer, número de dormitorios dobles.
- num\_duchas: integer, número de duchas.

De esta clase, distinguimos 3 subclases: restrictiva, preferida e indiferente, nótese que detonan el grado de obligatoriedad de la petición.

## **Cliente**

Clase que representa el usuario. Sus atributos son los siguientes:

- obtiene\_recomendaciones: instance, instancia de recomendaciones
- solicita\_peticion: instance, instancia de petición.

## **3.2. Metodología de resolución**

Existen diferentes técnicas de resolución de problemas que se pueden utilizar para las tareas que debe realizar un SCB. Nosotros utilizaremos la clasificación heurística ya que se adapta bien en nuestro caso, que tenemos un problema de análisis.

Hemos concluido que es un problema de análisis ya que sabemos el rango de las soluciones, que son básicamente instancias del sistema y por lo tanto solo necesitamos datos de la entrada para conseguir una salida/solución deseada.

El objetivo es obtener y representar el conocimiento necesario para que la asociación de datos personales y peticiones pueda realizarse y obtener una recomendación escogiendo una solución de entre un conjunto limitado.

La clasificación heurística se divide en tres fases: abstracción, asociación heurística y refinamiento. Iremos detallando cada una de estas fases.



## **1. Abstracción de datos**

Para abstraer los datos, partimos de las características que definen los usuarios especificando su importancia, es decir, si son restrictivas, preferentes o si les son indiferentes. Toda la información que no de el usuario, será almacenada para luego procesarla.

La realización de esta parte será mediante un conjunto de preguntas que hemos elaborado, si lo referenciamos a CLIPS, corresponde a la fase de hacer preguntas.

## **2. Asociación heurística**

Una vez extraída la información, es decir, pasar del problema concreto al problema abstracto, pasaremos el problema abstracto a la solución abstracta. Por ello, haremos una inferencia de las características de las viviendas que forman parte de la solución.

Como ya hemos mencionado anteriormente, con las peticiones que hace el cliente, distinguimos tres categorías: las restrictivas, las preferentes y las que son indiferentes. Entonces tenemos que primero se selecciona aquellas viviendas cuyas características cumplan las restricciones, una vez hecho esto, aumentamos la puntuación de la vivienda según la información del cliente.

El sistema de puntuación es bastante sencillo:

- 1) Primero sumaremos la puntuación según las características del cliente: Sumaremos 5 puntos si con cierta característica del cliente, la vivienda lo satisface y además lo ha marcado como obligatorio; sumaremos 3 puntos si con cierta característica la vivienda lo satisface y lo ha marcado como preferente; sumaremos 1 punto si con cierta característica, la vivienda lo satisface pero le es indiferente y finalmente también sumaremos 1 punto si no se cumple una característica del cliente pero la vivienda tiene algo que lo beneficie. Además de alacena el motivo por la cual aumentamos la puntuación.
- 2) Luego sumaremos la puntuación según las características de la vivienda. En este apartado no tratamos aquellas que son indiferentes. Si la obligatoriedad es preferente y las características indicadas por el cliente son preferentes, entonces sumamos 1 punto. Adicionalmente, si lo ha marcado como obligatorio, si el grado es adecuado, miramos si tiene algún detalle que lo haga destacar y si el grado es parcialmente adecuado almacenamos qué es lo que no ha cumplido.

Si relacionamos esta parte con CLIPS, corresponde con el módulo de abstracción.

### 3. Refinamiento

Finalmente, antes de imprimir la solución, ordenaremos las viviendas de manera que primero se imprima las mejores y luego las peores. Por ello tendremos tres grados diferentes: muy recomendable, adecuado y parcialmente adecuado. Además, en cada grado, el orden de las viviendas será definido por la puntuación. Por último, podremos mostrar las recomendaciones al usuario.

En el CLIPS, esta fase corresponde al módulo de solución.

## 4. Implementación

### 4.1. Construcción de la ontología

Para la construcción de la ontología, hemos utilizado Protegé, también hemos añadido una colección de instancias. Luego hemos exportado tanto las clases como las instancias a formato CLIPS generando así un fichero .clp donde escribiremos nuestro programa principal.

Además de las clases generadas usando Protegé, hemos creado una clase auxiliar llamada `ViviendaAbstracta` que nos será útil para representar las recomendaciones, es decir, la solución. Esta clase permite guardar la vivienda y otras particularidades de la vivienda como por ejemplo su grado de recomendación, por lo tanto nos será útil para tener accesible la información de las viviendas recomendadas.

Por último, también mencionar que hemos utilizado tres `deftemplates`. El primero se llama `Perfil` y este nos permite guardar información personal del cliente. El segundo se llama `Características` y básicamente representa la petición del cliente, por lo tanto contendrá información relacionada con las viviendas. Y el último template se llama `Obligatoriedad`, que nos permite saber si una característica es restrictiva, preferible o indiferente.

### 4.2. Módulos

En este apartado explicaremos los diferentes módulos que hemos implementado detallando su funcionalidad. En total tenemos 4 módulos, por orden de ejecución tenemos los siguientes módulos: “MAIN”, “PREGUNTAS” (dividido en preguntas al usuario y preguntas sobre las características), “ABSTRACCIÓN” y “SOLUCIÓN”.

### **Módulo “MAIN”**

Este es el módulo principal del sistema. Contiene una regla únicamente para presentar el programa y pasar al siguiente módulo, el de las preguntas.

### **Módulo “PREGUNTAS”**

Este módulo obtendrá los datos de entrada del usuario. Hace preguntas y lee respuestas a través de la línea de comandos. Hemos dividido este módulo en dos partes, la primera parte será un conjunto de preguntas sobre el usuario, es decir aspectos personales y la segunda parte será un conjunto de preguntas sobre las viviendas. En el caso de la segunda parte, también se le pedirá al usuario cual es el nivel de obligatoriedad de cada pregunta, esto es si lo quiere, lo prefiere o si le es indiferente.

Mientras vayamos leyendo los datos, modificaremos los template mencionados anteriormente hasta que se hayan rellenado completamente.

### **Módulo “ABSTRACCIÓN”**

Este módulo se encarga de considerar todas las viviendas existentes en el sistema y con la ayuda de los datos obtenidos en el módulo anterior, se hace una selección de las viviendas que serán la solución abstracta. La selección se hace mediante un filtrado, se descarta aquellos individuos que no cumplan un cierto número de restricciones que ha impuesto el usuario. Después, con las viviendas que hayamos seleccionado, determinaremos características sobre este, como por ejemplo su grado de recomendación, los servicios cercanos a él o su valoración. Recordemos que todo esto se encuentra en la clase que hemos definido nosotros mismos para poder representar la solución.

### **Módulo “SOLUCIÓN”**

Por último tenemos el módulo solución, que básicamente muestra una lista de viviendas según el grado de recomendación. Primero se mostrará las que son muy recomendables, luego las que son adecuadas y finalmente las que son parcialmente adecuadas de manera ordenada según la puntuación de la vivienda. Además en caso de que sea muy recomendable, también indicaremos qué características las hace así y en el caso de parcialmente adecuadas también indicaremos qué características las hace así.

### 4.3. Prototipos del problema

Antes de empezar a programar, nos planteamos encontrar una solución muy básica pero válida. El primer prototipo consiste solamente en filtrar aquellas viviendas que cumplieran el requisito del precio que indicaba el cliente.

Posteriormente, hemos ido extendiendo el prototipo a medida que nos surgían nuevas ideas respecto a los criterios de escoger una solución. Estas funcionalidades las íbamos añadiendo conforme ampliamos el proyecto.

#### Prototipo Final

En esta sección explicaremos el funcionamiento de nuestro prototipo final.

El sistema empezará por el módulo MAIN inicializando el programa con un bienvenida al usuario. Luego pasaremos al módulo de preguntas.

En el módulo de preguntas se harán una serie de preguntas ya vistas en la sección de desarrollo de la ontología. Primero preguntaremos sobre lo personal del cliente y luego haremos preguntas más focalizadas a las características de la vivienda indicando la obligatoriedad de esta: obligatorio/restrictivo, preferente o si le es indiferente.

Finalizada la fase de preguntas, entramos a la parte de abstracción. Es aquí donde seleccionaremos una parte de las instancias del sistema como solución abstracta. Para ello hacemos lo siguiente:

- 1) Seleccionamos todas las viviendas existentes dentro del sistema y comprobaremos si cumplen o no las restricciones/preferencias que ha indicado el usuario. En el caso donde el cliente sí que ha impuesto restricciones, iremos contando cuantas se incumplen para así poder asignarle un grado cualitativo. Nótese que en el caso de que no haya restricciones y por lo tanto solo hay preferencias e indiferencias, el grado cualitativo será adecuado.
- 2) Una vez finalizada la comprobación de las restricciones, pasamos a crear una solución abstracta, en esta solución se especifica la vivienda, los servicios cercanos a este, su grado cualitativo junto sus detalles, la puntuación junto sus detalles. **Nota importante:** para definir que una vivienda es muy recomendada, primero se evalúa si es adecuada, en tal caso, se mira si lo que tiene la vivienda es mejor que lo que ha especificado el usuario, si es el caso entonces será muy recomendado.

El sistema de puntuación lo hemos visto en el apartado de formalización.

Finalmente, pasamos al módulo solución donde imprimimos las viviendas según el grado de recomendación en orden descendente según la puntuación.

## 5. Juegos de prueba

En esta sección, hemos diseñado algunos juegos de prueba para verificar el funcionamiento de nuestro sistema. Para que la visualización sea amena, solo se imprimirá las 5 mejores viviendas en los tres grados de recomendación debido a que en el sistema disponemos de 151 instancias (viviendas + servicios).

**Nota:** Tanto el input como el output lo adjuntamos en ficheros.

### Prueba 1. Todo restrictivo (JP1.txt)

En esta prueba verificaremos el funcionamiento del sistema cuando el cliente indique que todas las características de la vivienda han de ser restrictivas ya sea de obligatoriedad positiva o negativa. Para ello hemos contestado a las preguntas de manera que encaje con las características de cierta vivienda. Nosotros hemos escogido la primera vivienda situada en Ciutat Vella ([ViviendaCV-1]).

El resultado es el que esperábamos. La información de la vivienda se encuentra en el fichero de las instancias. Pero básicamente, en la parte de las preguntas, estamos respondiendo según las características de la vivienda y por lo tanto, lógicamente la única vivienda que cumple con lo que ha indicado el usuario es [ViviendaCV-1].

El motivo por el cual el grado de recomendación es adecuado es debido a que estamos introduciendo los valores exactos que tiene y por lo tanto no hay características mejores de lo que pide el usuario.

Respecto a la puntuación de la vivienda, esta se ha calculado en base a la información personal del perfil. Vease el sistema de puntuación en el apartado de Formalización. Básicamente como que el usuario no tiene coche, entonces estaría bien recomendarle viviendas que tenga transporte cerca, pero como que ha marcado que lo quiere obligatoriamente y [ViviendaCV-1] tiene un servicio de transporte cerca, entonces se suma a la puntuación.

### Prueba 2. Todo preferente (JP2.txt)

En esta prueba veremos el comportamiento del sistema cuando el cliente no impone ninguna obligatoriedad, pero sí preferencias para todas las preguntas.

El resultado esperado sería todas las instancias de las viviendas, pero como ya hemos dicho solo veremos las 5 mejores viviendas.

Vemos que el listado de viviendas es adecuado, debido a que como hay restricciones y como que tampoco hay indiferencias, únicamente estamos ordenando las viviendas según las preferencias.

### **Prueba 3. Todo indiferente (JP3.txt)**

En esta prueba veremos el comportamiento del sistema cuando el cliente no impone ninguna obligatoriedad ni preferencia pero le es indiferente todas las características de la vivienda.

El resultado esperado sería todas las instancias de viviendas presentes en el sistema, nosotros, sin embargo, solo mostraremos 5 viviendas.

Vemos que el listado de viviendas es adecuado, debido a que no hay restricciones. Como tampoco se ha especificado ninguna preferencia, lo único que repercute en la puntuación de una vivienda son las características del perfil del cliente.

### **Prueba 4. Imposición sobre unifamiliares (JP4.txt)**

En esta prueba veremos el comportamiento de una ejecución donde el cliente imponga una restricción de que queremos recomendaciones sobre las viviendas unifamiliares.

Similar al caso anterior, podemos observar que hemos obtenido una lista de viviendas por cada grado de recomendación. En el caso de muy recomendado hay 2 viviendas donde podemos ver por qué motivos son muy recomendados; en el caso de adecuado solo observamos una vivienda y por último en el caso de las viviendas parcialmente adecuadas vemos 5 (que podrían ser más) junto con sus detalles del porqué este grado.

### **Prueba 5. Imposición sobre pisos (JP5.txt)**

En esta prueba veremos el comportamiento de una ejecución donde el cliente imponga una restricción de que queremos recomendaciones sobre los pisos.

Podemos observar que hemos obtenido una lista de viviendas por cada grado de recomendación. En el caso de muy recomendado hay 2 viviendas donde podemos ver por qué motivos son muy recomendados; en el caso de adecuado solo observamos una vivienda y por último en el caso de las viviendas parcialmente adecuadas 5 (que podrían ser más) junto con sus detalles del porqué este grado.

**Prueba 6. Imposición sobre duplex (JP6.txt)**

En esta prueba veremos el comportamiento de una ejecución donde el cliente imponga una restricción de que queremos recomendaciones sobre los duplex.

Se puede ver que hemos obtenido una lista de viviendas por cada grado de recomendación. En el caso de muy recomendado hay 1 vivienda donde podemos ver los detalles del por qué son muy recomendados; en el caso de adecuado no observamos ninguna y por último en el caso de las viviendas parcialmente adecuadas vemos 5 (que podrían ser más) junto con sus detalles del porqué este grado.

**Prueba 7. Cliente exigente (JP7.txt)**

Puede darse el caso de que el cliente sea muy exigente con las viviendas y que a pesar de que disponemos de una gran cantidad de viviendas en nuestro sistema, no hay ninguna vivienda que agrade al cliente. En esta prueba veremos este caso y veremos que el sistema no recomendará ninguna vivienda.

En la salida vemos efectivamente que “No se han encontrado viviendas que cumplan los criterios especificados”.

## **6. Conclusiones**

En este punto, podemos concluir que hemos sido capaces de cumplir con nuestros objetivos iniciales; dar respuesta a peticiones de los clientes de la forma más satisfactoria posible. No obstante, somos conscientes del potencial de mejora de este y cualquier sistema basado en el conocimiento, ya que cuanto mayor sea el conocimiento mejor será el resultado.

Mediante la realización de esta práctica hemos comprobado la efectividad y necesidad de los SBC con una de sus potenciales aplicaciones. Inicialmente, la cantidad de información necesaria para su ejecución y la conveniencia de la subdivisión en subproblemas ya suscitaron esta idea.



## 7. Planificación

Semana del 1 de Noviembre

- Ojeada al enunciado de la práctica, para conocer el contexto.

Semana del 8 y 15 de Noviembre

- Lectura de la práctica en su totalidad
- Extraer conceptos importantes para poder elaborar la ontología del sistema.
- Elaboración de la ontología: definición de las clases del problema, sus atributos y las relaciones entre ellos.

Semana del 22 de Noviembre

- Revisión de algunos conceptos de la ontología.
- Elaboración de un conjunto de preguntas para hacerle al usuario.
- Diseñar y planificar los diferentes módulos para resolver el problema principal y también diseñar los templates que se tendrá que usar.

Semana del 29 de Noviembre

- Empezar a implementar el sistema en CLIPS.
- Comenzar con una implementación sencilla de los diferentes módulos MAIN, PREGUNTAS, ABSTRACCIÓN y SOLUCIÓN
- Implementación definitiva del menú de bienvenida (MAIN).
- Implementación parcial de algunas preguntas (PREGUNTAS).

Semana del 6 de Diciembre

- Acabar el prototipo inicial con un filtrado sencillo.
- Una vez que teníamos el prototipo íbamos extendiendo según las necesidades que salían como por ejemplo, añadir algunas preguntas, añadir tratamiento de las puntuaciones de las viviendas, añadir más filtros.
- Empezar con la documentación.

Semana del 7 de Diciembre

- Acabar el prototipo final.
- Revisión del prototipo final y seguir con la documentación.
- Elaborar juegos de prueba y documentarlo.