

# Práctica de Sistemas Basados en el Conocimiento

## Recomendación de Viviendas en Alquiler

Grupo XX  
Nombres de los integrantes

1 de diciembre de 2025

### 1. Introducción y planteamiento del problema

#### 1.1. Estructura del documento

Este documento se organiza en tres partes principales, siguiendo las directrices de la rúbrica de evaluación:

- **Parte 1 (Introducción):** Planteamiento del problema, objetivos y metodología.
- **Parte 2 (Desarrollo):** Conceptualización, formalización (ontología) e implementación en CLIPS.
- **Parte 3 (Resultados):** Casos de prueba, análisis de resultados, conclusiones y trabajo en equipo.

#### 1.2. Contexto y motivación

La práctica plantea el desarrollo de un sistema de recomendación de viviendas en alquiler que pueda ser utilizado por una entidad pública para ayudar a los ciudadanos a encontrar ofertas adecuadas a sus necesidades.

El problema implica combinar dos tipos de información:

- **Características de las viviendas:** precio, superficie, tipo de propiedad, amenities, ubicación, etc.
- **Perfil y preferencias del usuario:** presupuesto, composición familiar, necesidades de proximidad a servicios, restricciones, etc.

La complejidad del dominio y la necesidad de aplicar conocimiento experto para ponderar múltiples factores convierten este problema en un caso idóneo para un Sistema Basado en el Conocimiento (SBC).

### 1.3. Objetivos de la práctica

El objetivo principal de esta práctica es diseñar e implementar un **Sistema Basado en el Conocimiento (SBC)** que emule el razonamiento de un experto inmobiliario para recomendar viviendas en alquiler. Los objetivos específicos son:

- Analizar el dominio del problema e identificar las fuentes de conocimiento necesarias.
- Construir una ontología formal que represente los conceptos clave: **viviendas, solicitantes, servicios urbanos y zonas de la ciudad**.
- Implementar en CLIPS un sistema de reglas que realice:
  - Evaluación de compatibilidad entre perfil del solicitante y características de la vivienda.
  - Cálculo de proximidad a servicios relevantes (transporte, educación, ocio, salud, etc.).
  - Clasificación de las recomendaciones en tres categorías: *parcialmente adecuada, adecuada y muy recomendable*.
- Validar el sistema mediante casos de prueba representativos y documentar el proceso de desarrollo siguiendo la metodología de ingeniería del conocimiento.

### 1.4. Metodología de desarrollo

Para abordar este problema, se ha seguido la metodología de **Ingeniería del Conocimiento** explicada en clase, estructurada en las siguientes fases:

1. **Identificación:** Análisis del enunciado, definición de objetivos y alcance.
2. **Conceptualización:** Extracción y organización del conocimiento experto, identificación de conceptos y relaciones.
3. **Formalización:** Construcción de la ontología en Protégé y diseño de la estrategia de razonamiento.
4. **Implementación:** Codificación en CLIPS del sistema de reglas y hechos.
5. **Prueba:** Validación mediante casos de prueba y análisis de resultados.

## 2. Desarrollo de la solución

### 2.1. Conceptualización

#### 2.1.1. Extracción de conocimiento

Si se usó modelo de lenguaje: contexto dado, preguntas realizadas, conocimiento extraído. Si no, explicar cómo se obtuvo el conocimiento (sentido común, investigación).

### 2.1.2. Conceptos y relaciones principales

A partir del análisis del dominio y de la experiencia en portales inmobiliarios, identificamos los siguientes conceptos clave, que posteriormente se formalizarán en la ontología:

- **Cliente**: persona o grupo que busca vivienda. Se distingue entre:
  - **Individual**: Adulto, Joven, Estudiante, Anciano.
  - **Grupo**: Pareja (con o sin hijos previstos, joven o anciana), Familia (con o sin ancianos), Grupo de estudiantes.
- **Vivienda**: oferta de alquiler. Incluye subtipos como:
  - **Piso, Dúplex, Ático, Estudio**.
  - **Casa individual, Casa pareada, Casa adosada**.
  - **Habitación** para alquiler individual.
- **Servicio**: elemento urbano que influye en la calidad de vida:
  - **Transporte**: Parada de autobús, metro, tren, aparcamiento.
  - **Educación**: Colegio, instituto, universidad.
  - **Salud**: Clínica, hospital, farmacia.
  - **Comercio**: Supermercado, hipermercado, centro comercial, tienda.
  - **Ocio**: Parque, piscina, gimnasio, cine, museo, zona de vida nocturna, restaurante, playa, polideportivo.
- **Zona**: área de la ciudad con características homogéneas:
  - **Residencial, Centro, Núcleo urbano, Negocios, Industrial**.

#### Relaciones principales identificadas:

- Una **Vivienda** *está ubicada en* una **Zona** (relación `is_located_in_zone`).
- Una **Vivienda** *está cerca de* uno o más **Servicios** (relación `is_near_service`).
- Un **Cliente** *tiene preferencias y restricciones* sobre:
  - Precio máximo, flexibilidad presupuestaria, precio mínimo sospechoso (*too bargain*).
  - Número mínimo de dormitorios y baños.
  - Necesidad de transporte público, posesión de coche.
  - Ubicación de trabajo/estudio (coordenadas).
  - Mascotas, ruido tolerado, características deseadas (terraza, ascensor, amueblado, etc.).
- Una **Vivienda** *puede satisfacer o no* dichas preferencias, lo que determina su **grado de recomendación** (parcialmente adecuada, adecuada, muy recomendable).

### 2.1.3. Descomposición en subproblemas

Explicar cómo se divide el problema global en tareas más pequeñas:

- Subproblema 1: Captura de preferencias del usuario.
- Subproblema 2: Evaluación de viviendas respecto a preferencias.
- Subproblema 3: Cálculo de distancias a servicios.
- Subproblema 4: Clasificación y recomendación final.

## 2.2. Formalización

### 2.2.1. Ontología del dominio

La ontología se ha construido en Protégé siguiendo la metodología de desarrollo de ontologías 101, y está compuesta por 4 conceptos raíz principales, sus jerarquías de subclases, propiedades de objeto y datos, y restricciones de dominio/rango.

#### Jerarquía de clases

- **Client\_Group**: representa a los solicitantes. Se divide en:
  - Individual: Adult, Young, Student, Elderly.
  - Group: Couple (Elderly\_Couple, Planning\_Kids, Young\_No\_Kids), Family (No\_Elderly, With\_Elderly), Student\_Group.

Se ha modelado así para capturar perfiles con necesidades diferenciadas (ej: jóvenes buscan ocio, familias buscan escuelas).

- **Property**: representa las viviendas en alquiler. Incluye:
  - Tipos urbanos: Apartment, Duplex, Penthouse, Studio, Triplex, Room.
  - Tipos unifamiliares: Detached\_House, Semidetached\_House, Row\_House, Single-family\_House.

Se han definido como *disjoint* para evitar inconsistencias.

- **Service**: modela los servicios urbanos. Subclases:
  - Transport (Bus\_stop, Metro\_station, Train\_station, Parking, Cable\_car\_station).
  - Education (School, High\_School, University).
  - Healthcare (Clinic, Hospital, Pharmacy).
  - Shopping (Supermarket, Hipermarket, Mall, Store).
  - Recreation (Park, Pool, Gym, Beach, Cinema, Museum, Nightlife, etc.).

Esta categorización permite evaluar la proximidad a servicios de forma estructurada.

- **Zone**: describe zonas de la ciudad:
  - Residential, Business, Downtown, Industrial, Urban\_core.

Cada zona tiene atributos como nivel de ruido y seguridad.

## Propiedades de objeto

- `is_located_in_zone`: relaciona una `Property` con una `Zone`. Es funcional (cada vivienda está en una sola zona).
- `is_near_service`: relaciona una `Property` con un `Service`. Permite modelar proximidad a múltiples servicios.

## Propiedades de datos

Se han agrupado en tres categorías para facilitar la modularidad:

- `property_attribute`: atributos de la vivienda (precio, metros<sup>2</sup>, habitaciones, terraza, ascensor, amueblado, etc.).
- `client_attribute`: atributos del solicitante (edad, presupuesto, preferencias, ubicación trabajo/estudio, etc.).
- `zone_attribute`: atributos de la zona (nivel de ruido, seguridad).

Se han usado tipos de datos específicos (`boolean`, `int`, `float`) y enumerados (ej: `Sun_Time`: `Morning`, `Afternoon`, `All_Day`, `Never`).

## Decisiones de diseño justificadas

- **Coordenadas enteras**: se usan `geo_lat` y `geo_long` como enteros para simplificar el cálculo de distancias en CLIPS.
- `floor_level` solo aplica a `Apartment`, `Room` y `Studio`, no a viviendas unifamiliares.
- `Too_Bargain`: se incluye como atributo del cliente para detectar precios sospechosamente bajos.
- *Disjoint* entre clases: evita que una misma instancia sea, por ejemplo, un `Apartment` y una `Detached_House`.

### 2.2.2. Metodología de resolución de problemas

Explicar qué metodología se ha elegido (p. ej., clasificación, razonamiento basado en casos) y cómo los subproblemas identificados encajan en ella.

### 2.2.3. Representación en reglas

Descripción de cómo se traduce el conocimiento en reglas de producción. Estructura general del sistema de reglas (módulos, estrategia de control).

## 2.3. Implementación en CLIPS

### 2.3.1. Estructura del proyecto

Organización de módulos, templates, hechos iniciales.

### 2.3.2. Ejemplo de reglas clave

Fragmentos de código comentados que ilustren el razonamiento.

```
(defrule evaluar-precio
  ?s <- (solicitante (precio-max ?pmax) (flexible ?flex))
  ?v <- (vivienda (precio ?p) (id ?id))
  (test (<= ?p ?pmax))
  =>
  (assert (candidata (id ?id) (motivo "Dentro de presupuesto"))))
```

Listing 1: Ejemplo de regla en CLIPS

### 2.3.3. Desarrollo incremental

Descripción de los prototipos desarrollados, evolución desde el primer prototipo funcional hasta la versión final.

## 3. Resultados y validación

### 3.1. Juegos de prueba

Explicación de cómo se seleccionaron los casos: perfiles variados (estudiante, familia, pareja), casos límite, cobertura de funcionalidades.

### 3.2. Ejecución de pruebas

Tabla resumen con al menos 6 casos diferentes, mostrando:

- Perfil de entrada.
- Viviendas recomendadas (adecuadas, parcialmente adecuadas, muy recomendables).
- Criterios incumplidos o características destacadas.

Cuadro 1: Ejemplo de casos de prueba

Perfil	Vivienda	Grado	Observaciones
Estudiante, precio bajo	Atico céntrico	Parcial	Sin terraza
Familia con hijos	Dúplex con jardín	Muy recomendable	Cerca de colegio y parque

### 3.3. Análisis de resultados

Explicación de por qué el sistema toma ciertas decisiones, ejemplos concretos basados en el conocimiento codificado.

## **3.4. Conclusiones**

### **3.4.1. Logros y dificultades**

Qué se ha conseguido, qué partes fueron más complejas, limitaciones del sistema.

### **3.4.2. Trabajo en equipo**

Breve descripción de cómo se organizó el grupo, reuniones, reparto de tareas, uso de la planificación propuesta.

### **3.4.3. Posibles mejoras**

Extensiones futuras: integración con datos reales de mapas, más factores de ponderación, interfaz gráfica.

## **A. Anexo A: Ontología completa**

Gráfico exportado desde Protégé con la jerarquía de clases y propiedades.

## **B. Anexo B: Diálogo con modelo de lenguaje**

Si aplica: contexto usado, preguntas más relevantes y respuestas resumidas.

## **C. Anexo C: Código fuente completo**

Listado completo del código CLIPS (o enlace al repositorio).