

---

# PRÁCTICA 4

## Representación de conocimiento de alto nivel usando Protégé

---

### INGENIERÍA DEL CONOCIMIENTO

Hossam El Amraoui Leghzali  
26533841Y  
hossam@correo.ugr.es

Curso 2023/2024  
Grupo 3 (Martes 17:30 - 19:30)  
Grado en Ingeniería Informática  
Universidad de Granada

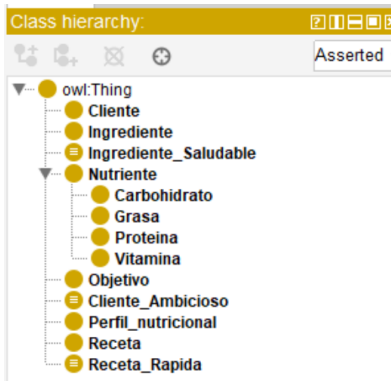


**UNIVERSIDAD  
DE GRANADA**

## 1. Implementación de los diferentes elementos de la ontología

Se han implementado en la práctica los elementos pedidos. A continuación, se muestra una imagen de cada una de las partes de la ontología implementadas:

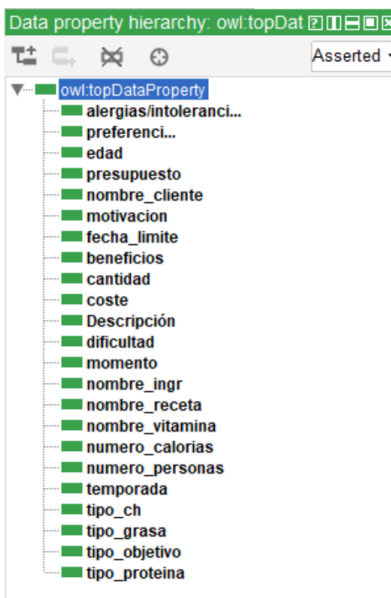
### ■ Clases:



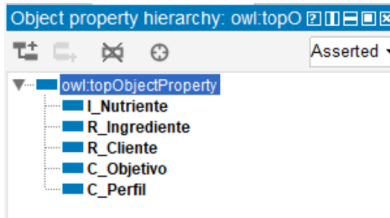
Aparte de las clases primitivas, se han definido 3 clases a partir de ellas:

- Ingrediente\_Saludable
- Cliente\_Ambicioso
- Receta\_Rapida

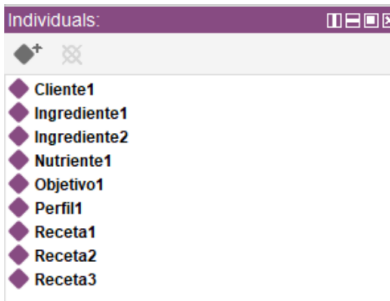
### ■ Propiedades de tipo dato:



- Propiedades de tipo objeto:

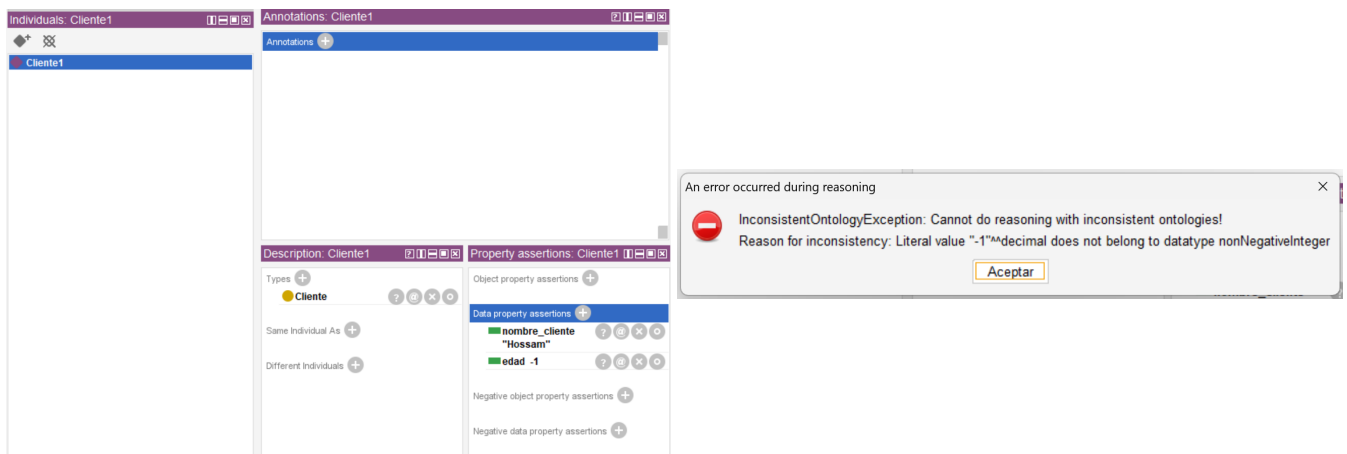


- Instancias:



## 2. Uso del razonador

A medida que se iba haciendo la práctica, se ha ido aplicando el razonador para comprobar que la ontología estaba bien definida y que no había fallos de consistencia. En la siguiente imagen se muestra el resultado de la aplicación del razonador en un error intencionado (la edad del cliente debe ser un número positivo, pero se le ha introducido un número negativo):



### 3. Ejercicios de la práctica

#### 3.1. 1 ejemplo de valor deducido por el razonador y la explicación dada

The screenshot displays a reasoning interface with four main panels: **Individuals: Nutriente1**, **Annotations: Nutriente1**, **Description: Nutriente1**, and **Property assertions: Nutriente1**.

- Individuals: Nutriente1**: A list of individuals including **Cliente1**, **Ingrediente1**, **Ingrediente2**, **Nutriente1** (highlighted), **Objetivo1**, **Perfil1**, **Receta1**, **Receta2**, and **Receta3**.
- Annotations: Nutriente1**: A panel for annotations, currently empty.
- Description: Nutriente1**: A panel showing the type **Proteina** (highlighted) and options for **Same Individual As** and **Different Individuals**.
- Property assertions: Nutriente1**: A panel showing property assertions. Under **Data property assertions**, there is an assertion: **tipo\_proteina "Vegetal"**.

An **Explanation** dialog box is open, titled "Explanation for Nutriente1 Type Proteina". It contains the following options:

- ☒ Show regular justifications
- ☒ All justifications
- ☐ Show laconic justifications
- ☐ Limit justifications to 2

The explanation shows:

Explanation 1 ☐ Display laconic explanation

Explanation for: Nutriente1 Type Proteina

- Nutriente1 tipo\_proteina "Vegetal"
- tipo\_proteina Domain Proteina

Se puede observar que el razonador ha deducido que **Nutriente1** es una instancia de la clase **Proteina** debido a que **Nutriente1** tiene como atributo **"tipo\_proteina"**, cuyo dominio es la clase **Proteina**.

### 3.2. 1 ejemplo de axioma de una clase deducido por el razonador y la explicación dada

The screenshot displays a web-based ontology editor interface. On the left, the 'Class hierarchy: Receta\_Rapida' panel shows a tree structure starting from 'owl:Thing', with 'Receta\_Rapida' highlighted at the bottom. The main area is divided into two panels: 'Annotations: Receta\_Rapida' and 'Description: Receta\_Rapida'. The 'Annotations' panel shows an 'rdfs:label' annotation for 'Receta\_Rapida'. The 'Description' panel shows the class description for 'Receta\_Rapida' as '(R\_Ingrediente some Ingrediente) and (R\_Ingrediente max 5 Ingrediente)'. Below this, it shows 'Receta' as a subclass of 'Receta\_Rapida'. At the bottom, a pop-up window titled 'Explanation for Receta\_Rapida SubClassOf (R\_Ingrediente some Ingrediente) and (R\_Ingrediente min 2 Ingrediente)' provides a detailed explanation of the deduction. It shows that 'Receta\_Rapida' is equivalent to '(R\_Ingrediente some Ingrediente) and (R\_Ingrediente max 5 Ingrediente)', and 'Receta' is equivalent to '(R\_Ingrediente some Ingrediente) and (R\_Ingrediente min 2 Ingrediente)'. The explanation also shows that 'Receta' is a subclass of 'Receta\_Rapida'.

A la clase **Receta** se le ha añadido un axioma que indica que una receta debe tener por lo menos 2 ingredientes. En base a eso, como se puede observar en la imagen, el razonador ha deducido que la clase **Receta\_Rapida** también debe tener ese axioma debido a que es una subclase de la clase **Receta**, cosa que ocurre por la razón que se explicará en el siguiente apartado.

### 3.3. 1 ejemplo de relación de jerarquía deducida por el razonador y la explicación dada

The screenshot displays a software interface with three main panels and an explanation window.

- Class hierarchy: Cliente\_Ambicioso**: A tree view showing a hierarchy of classes. The root is `owl:Thing`, followed by `Cliente`, `Ingrediente`, `Ingrediente_Saludable`, `Nutriente`, and `Objetivo`. Below `Objetivo` are `Carbohidrato`, `Grasa`, `Proteina`, and `Vitamina`. The class `Cliente_Ambicioso` is highlighted in blue, indicating it is the selected class.
- Annotations: Cliente\_Ambicioso**: A panel showing annotations for the selected class. It displays `rdfs:label` with the value `Cliente_Ambicioso`.
- Description: Cliente\_Ambicioso**: A panel showing the description of the selected class. It includes:
  - Equivalent To**: `(C_Objetivo some Objetivo) and (C_Objetivo min 5 Objetivo)`
  - SubClass Of**: `Cliente`
  - General class axioms**: A section for general class axioms.
  - SubClass Of (Anonymous Ancestor)**: A section for sub-class relationships with anonymous ancestors.
- Explanation for Cliente\_Ambicioso SubClassOf Cliente**: A dialog box showing the justification for the deduced sub-class relationship. It includes:
  - Buttons for **Show regular justifications**, **Show laconic justifications**, **All justifications**, and **Limit justifications to** (set to 2).
  - Explanation 1**: A section showing the explanation for the sub-class relationship. It includes the text `Cliente_Ambicioso EquivalentTo (C_Objetivo some Objetivo) and (C_Objetivo min 5 Objetivo)` and `C_Objetivo Domain Cliente`.

A las 3 clases añadidas en la práctica se les ha añadido un axioma que justifique la creación de la clase en base a las clases primitivas. En este ejemplo, la clase **Cliente\_Ambicioso** tiene un axioma que indica que un cliente es ambicioso si tiene por lo menos 5 objetivos. Como esta clase contiene la propiedad de tipo objeto **C\_Objetivo**, el razonador ha deducido que la clase **Cliente\_Ambicioso** es una subclase de la clase **Cliente**. Lo mismo ocurre con las otras dos clases añadidas.