Practica 1: Sistemas expertos basados en reglas

Sistema para la Recomendación Personalizada de Series y Películas

Desarrollado por:

Andres Felipe Garcia Orrego

Carolina Humanez Urrego

Diego Fernando Juajibioy Chavez

Universidad Nacional de Colombia

Introducción a inteligencia artificial

2025-1

Contenido

[Descripción general 3](#_Toc197998150)

[Video pitch 4](#_Toc197998151)

[Descripción por componentes 4](#_Toc197998152)

[Sistema experta (Experta) 4](#_Toc197998153)

[Definición de hechos 4](#_Toc197998154)

[Reglas 5](#_Toc197998155)

[Uso de Prioridades (salience) 5](#_Toc197998156)

[Lógica Difusa (Scikit-Fuzzy) 5](#_Toc197998157)

[Definición del Universo y Variables 5](#_Toc197998158)

[Definición de Funciones de Pertenencia 5](#_Toc197998159)

[Graficas de funciones de pertenencia 7](#_Toc197998160)

[Reglas del Sistema 9](#_Toc197998161)

[Proceso de defuzzificación 9](#_Toc197998162)

[Ontología (RDFLib + OWL-RL) 10](#_Toc197998163)

[Clases 10](#_Toc197998164)

[Propiedades 10](#_Toc197998165)

[Diagrama de clases 3](#_Toc197998166)

[Casos de razonamiento 3](#_Toc197998167)

[Estructura (Grafo) 6](#_Toc197998168)

[Descripción De La Integración 7](#_Toc197998169)

[1. Interfaz con el usuario (Streamlit) 7](#_Toc197998170)

[2. Sistema de lógica difusa (Scikit-Fuzzy) 7](#_Toc197998171)

[3. Ontología (RDF/RDFS con RDFlib) 7](#_Toc197998172)

[4. Sistema experto (Experta) 8](#_Toc197998173)

[5. Presentación de resultados 8](#_Toc197998174)

# Descripción general

En un entorno digital donde el contenido audiovisual abunda, recomendar la serie o película adecuada a cada usuario se ha convertido en un reto clave. Ya no basta con basarse en calificaciones o coincidencias de género; los usuarios buscan recomendaciones que se alineen con su nivel de interés, su disposición a descubrir nuevos géneros y su perfil de edad. Por este motivo, este proyecto propone un **sistema inteligente híbrido** que integra un **sistema experto**, una **ontología semántica** y un **motor de inferencia difuso**, con el objetivo de generar recomendaciones más precisas, humanas y adaptadas al contexto del usuario.

Este sistema híbrido procesa entradas sobre las preferencias del usuario y las características del contenido para producir un nivel de recomendabilidad personalizado, expresado como un valor entre 0 y 100. La lógica difusa, en particular, permite modelar la subjetividad humana en la toma de decisiones, evaluando de forma gradual los factores más relevantes para la recomendación: el **interés del usuario**, su **apertura a nuevos géneros** y la **adecuación por edad**.

# Video pitch

##AQUÍ LINK

# Descripción por componentes

## Sistema experta (Experta)

### Definición de hechos

* **Película** (nombre , genero , clasificación , idioma , formato )

*Representa una película con sus atributos*

* **Serie** (nombre , genero , clasificación , idioma , formato )

*Representa una serie con sus atributos*

* **Usuario** (genero, idioma, formato, clasificación)

*Representa los gustos del usuario con respecto al género, clasificación, idioma y formato*

* **Preferencia** (**Tipo**)

*Representa el tipo de contenido que prefiere el usuario: película o serie*

* **Nivel de recomendación** (nivel)

*Representa el nivel de recomendabilidad del usuario (se calcula con lógica difusa)*

* **Plataforma preferida** (plataforma)

*Representa la plataforma preferida del usuario*

Para esta implementación particular:

**Valores para clasificación:** ["+7", "+16", "+18"]

**Valores para formato:** ["HD", "SD", "4K", "BluRay", "Streaming"]

**Valores para género:** ["CienciaFiccion", "Comedia", "Terror", "Drama", "Documental"]

### Reglas

###

### Uso de Prioridades (salience)

###

## Lógica Difusa (Scikit-Fuzzy)

### Definición del Universo y Variables

* **Interés del usuario**: valores de 0 a 10, incrementos de 0.1.
* **Apertura a nuevos géneros**: valores de 0 a 10, incrementos de 0.1.
* **Edad**: valores de 0 a 60, incrementos de 1.
* **Nivel de recomendabilidad (salida)**: valores de 0 a 100, incrementos de 1.

### Definición de Funciones de Pertenencia

#### **Interés del Usuario**

* **Bajo**: Triangular (0, 0, 4)
* **Medio**: Trapezoidal (3.5, 4.5, 5.5, 6.5)
* **Alto**: Gaussiana (media=8, sigma=1.2)
* **Muy interesado**: Concentración de **Alto** (potencia = 2)

#### **Apertura a Nuevos Géneros**

* **Cerrado**: Trapezoidal (0, 0, 2.5, 4)
* **Neutral**: Gaussiana (media=5, sigma=1.5)
* **Abierto**: Triangular (6, 8, 10)
* **Mas o menos abierto**: Dilatación de **Abierto** (potencia = 1.3)

#### **Adecuación por Edad**

* **Niño**: Triangular (0, 0, 12)
* **Adolescente**: Trapezoidal (10, 13, 17, 20)
* **Adulto**: Gaussiana (media=30, sigma=10)

#### **Nivel de Recomendabilidad**

* **Baja**: Trapezoidal (0, 0, 20, 35)
* **Media**: Gaussiana (media=50, sigma=10)
* **Alta**: Trapezoidal (65, 80, 100, 100)

### Graficas de funciones de pertenenciaGráfico, Gráfico de líneas El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

### Reglas del Sistema

1. **SI** interés del usuario es **alto** **Y** apertura a nuevos géneros es **abierto**, **ENTONCES** recomendabilidad es **alta**.
2. **SI** interés del usuario es **medio** **Y** apertura es **neutral**, **ENTONCES** recomendabilidad es **media**.
3. **SI** interés del usuario es **bajo** **O** apertura es **cerrado**, **ENTONCES** recomendabilidad es **baja**.
4. **SI** interés del usuario es **muy interesado** **Y** apertura es **mas o menos abierto**, **ENTONCES** recomendabilidad es **alta**.
5. **SI** apertura a nuevos géneros es **neutral** **Y** NO (edad es **niño**), **ENTONCES** recomendabilidad es **media**.
6. **SI** interés del usuario es **medio** **Y** edad es **adulto**, **ENTONCES** recomendabilidad es **media**.
7. **SI** edad es **niño** **Y** NO (apertura es **abierto**), **ENTONCES** recomendabilidad es **baja**.
8. **SI** interés es **alto** **O** edad es **adolescente**, **ENTONCES** recomendabilidad es **media**.
9. **SI** interés es **muy interesado** **Y** apertura es **cerrado** **Y** edad es **adulto**, **ENTONCES** recomendabilidad es **media**.

### Proceso de defuzzificación

Se utilizó el método de defuzzificación del centroide porque proporciona un valor de salida representativo y estable, al considerar toda el área bajo la curva de la función de pertenencia. Esto permite obtener una recomendación más precisa y equilibrada, siendo el método más común y confiable en sistemas de lógica difusa.

## Ontología (RDFLib + OWL-RL)

### Clases

* Contenido
* Serie, subclase de Contenido
* Película, subclase de Contenido
* Genero
* Director
* Actor
* Idioma
* Formato
* ClasificacionEdad
* Plataforma
* SerieAnimada, subclase de Serie
* PeliculaAnimada, subclase de Película
* IdiomaDoblado, subclase de idioma
* IdiomaOriginal, subclase de idioma

### Propiedades

**Propiedad 1: tieneGenero**

Establece el género al que pertenece un contenido audiovisual (serie o película).

**Dominio**: Contenido

**Rango**: Genero

**Tipo**: Objeto (URI)

**Propiedad 2: tieneDirector**

Relaciona un contenido con la persona que lo dirigió.

**Dominio**: Contenido

**Rango**: Director

**Tipo**: Objeto (URI)

**Propiedad 3: tieneActor**

Asocia un actor principal con una serie o película.

**Dominio**: Contenido

**Rango**: Actor

**Tipo**: Objeto (URI)

**Propiedad 4: tieneIdioma**

Indica el idioma original del contenido.

**Dominio**: Contenido

**Rango**: Idioma

Jerarquía: Subpropiedad de dcterms:language (Dublin Core)

**Tipo**: Objeto (URI)

Uso de vocabulario externo: Sí

**Propiedad 5: tieneFormato**

Define el formato físico o digital en que está disponible el contenido.

**Dominio**: Contenido

**Rango**: Formato

**Tipo**: Objeto (URI)

**Propiedad 6: tieneDuracion**

Representa la duración del contenido en minutos.

**Dominio**: Contenido

**Rango**: xsd:int

**Tipo**: Literal tipado

**Propiedad 7: tienePuntuacion**

Registra una calificación numérica (por ejemplo, entre 1 y 10) sobre la calidad del contenido.

**Dominio**: Contenido

**Rango**: xsd:float

**Tipo**: Literal tipado

**Propiedad 8: tieneEpisodios**

Indica cuántos episodios tiene una serie.

**Dominio**: Serie

**Rango**: xsd:int

**Tipo**: Literal tipado

**Propiedad 9: tieneClasificacion**

Especifica la clasificación por edad recomendada para el contenido.

**Dominio**: Contenido

**Rango**: ClasificacionEdad

**Tipo**: Objeto (URI)

**Propiedad 10: relacionContenido**

Relaciona dos contenidos entre sí, por ejemplo, cuando una película tiene una secuela o una serie relacionada.

**Dominio**: Contenido

**Rango**: Contenido

**Tipo**: Objeto (URI)

**Propiedad 11: disponibleEn**

Muestra en qué plataforma está disponible un contenido (por ejemplo, Netflix, HBO).

**Dominio**: Contenido

**Rango**: Plataforma

**Tipo**: Objeto (URI)

**Propiedad 12: basadaEnPelicula**

Indica que una serie se basa en una película existente.

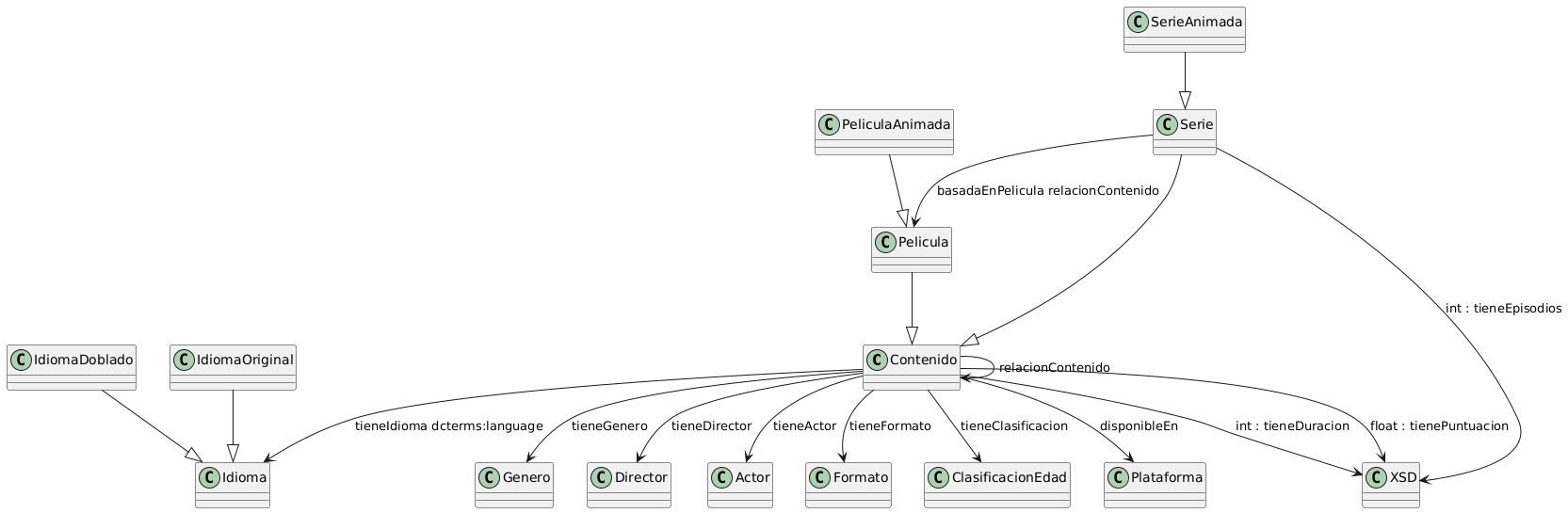
**Dominio**: Serie

**Rango**: Pelicula

**Tipo**: Objeto (URI)

*Jerarquía*: Subpropiedad de relacionContenido

### Diagrama de clases



### Casos de razonamiento

Para ver las inferencias el Código, dirigirse al apartado Instancias del archivo ontologia.py

#### Caso 1: Jerarquía de clases

1.Inferencia de Contenido mediante Pelicula:

En la ontología definida, se estableció explícitamente que la clase Pelicula es una subclase de Contenido:

:Pelicula rdfs:subClassOf :Contenido .

Esto significa que toda instancia de Pelicula también es, por herencia, una instancia de Contenido, aunque no se declare de forma directa.

Por ejemplo, al declarar que :inception es una instancia de Pelicula:

:inception rdf:type :Pelicula .

el razonador inferirá automáticamente el siguiente hecho adicional:

:inception rdf:type :Contenido .

Este nuevo hecho no estaba explícito en el grafo, pero fue deducido por el razonador gracias a la jerarquía de clases. Así, se reconoce que la película también es un tipo de contenido, lo cual puede ser útil para consultas más generales sobre obras audiovisuales sin distinguir si son series o películas.

2.Inferencia de Serie mediante Serie Animada:

De forma análoga, se definió en la ontología que SerieAnimada es una subclase de Serie:

:SerieAnimada rdfs:subClassOf :Serie .

Esto indica que cualquier recurso que sea instancia de SerieAnimada también será, por jerarquía, considerado automáticamente como una instancia de Serie.

Por ejemplo, si declaramos:

:bojackHorseman rdf:type :SerieAnimada .

el razonador RDFS deducirá automáticamente:

:bojackHorseman rdf:type :Serie .

Este razonamiento permite aprovechar las propiedades comunes definidas para Serie (como tieneEpisodios o disponibleEn), incluso si una serie ha sido clasificada como una variante más específica como lo es una serie animada.

#### Caso2: Miembro de una clase desde el dominio o rango de sus propiedades.

En la ontología, se ha definido que la propiedad :tieneEpisodios tiene como dominio la clase Serie, lo que significa que cualquier recurso que utilice esta propiedad es automáticamente considerado una instancia de Serie, según la semántica de RDFS.

:tieneEpisodios rdf:type rdf:Property .

:tieneEpisodios rdfs:domain :Serie .

:tieneEpisodios rdfs:range xsd:int .

Por ejemplo, si en el grafo RDF se incluye el siguiente hecho explícito:

:dark :tieneEpisodios 26 .

aunque no se indique directamente que :dark es una Serie, el razonador RDFS puede inferirlo gracias al dominio de la propiedad :tieneEpisodios:

:dark rdf:type :Serie .

Esta inferencia se conoce como razonamiento por dominio de propiedad, y permite deducir la clase de una entidad basándose únicamente en el uso de una propiedad cuya definición ya impone el tipo esperado de sujeto.

Este mecanismo es útil cuando se trabaja con datos parcialmente anotados, permitiendo enriquecer automáticamente el grafo con información coherente.

#### Caso 3: Relación subproperty

En la ontología, la propiedad :basadaEnPelicula está definida como una subpropiedad de :relacionContenido. Esto significa que cada vez que se usa :basadaEnPelicula, también se está utilizando implícitamente la propiedad más general :relacionContenido. Esta inferencia es posible gracias a la semántica RDFS y se conoce como razonamiento por subpropiedad.

:basadaEnPelicula rdf:type rdf:Property .

:basadaEnPelicula rdfs:domain :Serie .

:basadaEnPelicula rdfs:range :Pelicula .

:basadaEnPelicula rdfs:subPropertyOf :relacionContenido .

Por ejemplo, si se incluye el siguiente hecho en el grafo:

:arcane :basadaEnPelicula :league\_of\_legends .

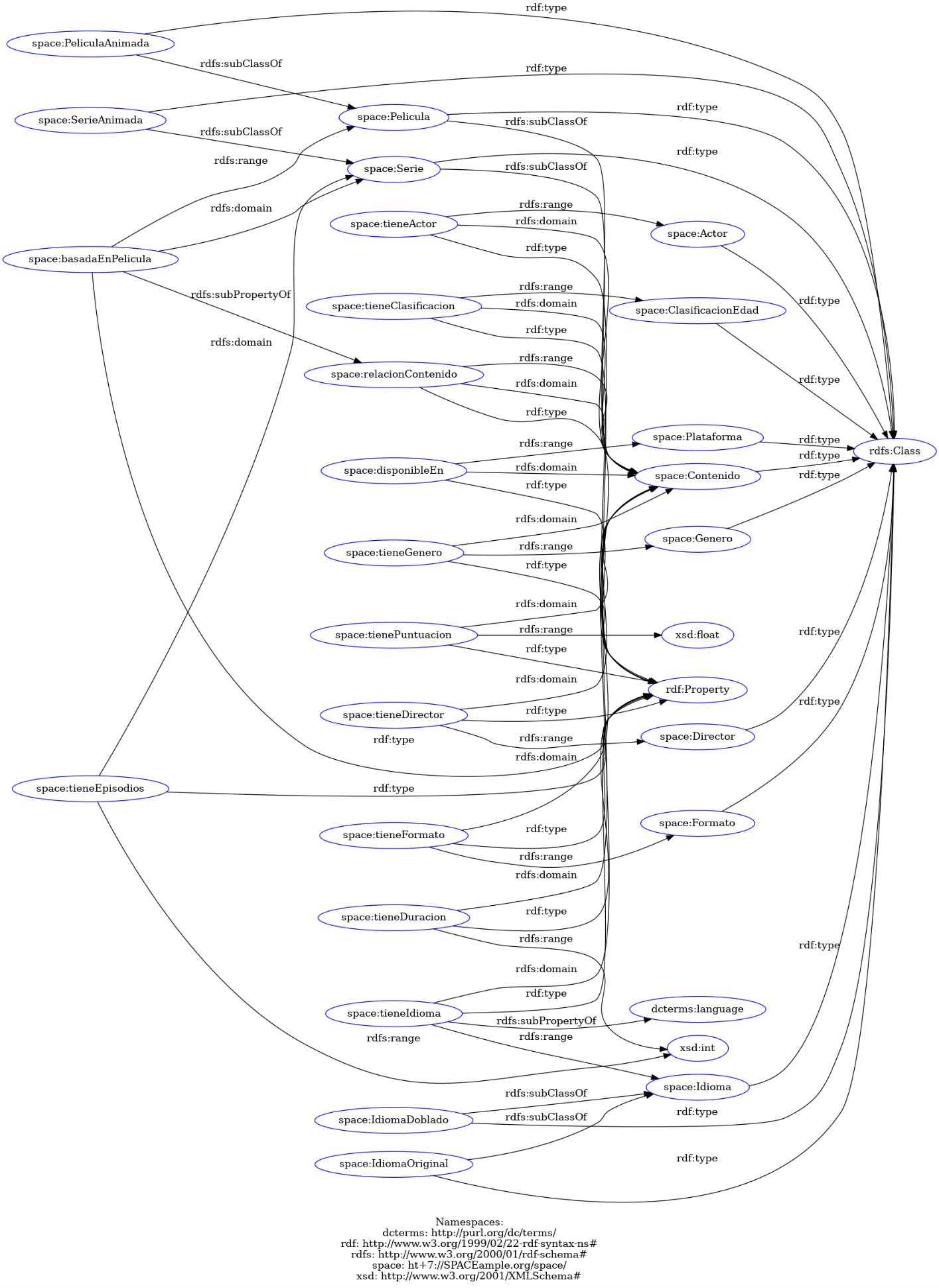
El razonador puede deducir automáticamente el hecho más general

:arcane :relacionContenido :league\_of\_legends .

Esto permite estructurar la ontología de forma más modular, definiendo propiedades específicas sin perder la capacidad de consultar o inferir relaciones más generales.

##### Nota: La comparación de los grafos antes y después del razonamiento se hizo en forma de código en el apartado de COMPARACIÓN

### Estructura (Grafo)



## Descripción De La Integración

Este sistema inteligente de recomendación de películas y series combina tres enfoques: una **ontología en RDF/RDFS**, un **sistema de lógica difusa** y un **motor de reglas experto**, integrados en una interfaz construida con la librería **Streamlit**.

### 1. Interfaz con el usuario (Streamlit)

La aplicación comienza solicitando al usuario que complete un formulario con sus preferencias:

* Nombre, edad, género favorito, idioma preferido.
* Formato deseado (película o serie) y plataforma favorita (Netflix, DisneyPlus, HBO\_Max).
* Además, el usuario indica dos valores numéricos: **nivel de interés** y **apertura a nuevos géneros**, que se ingresan mediante deslizadores.

### 2. Sistema de lógica difusa (Scikit-Fuzzy)

Con los valores proporcionados, se activa un sistema de lógica difusa para evaluar la **recomendabilidad** del usuario.

* Se utiliza un sistema de control difuso (ControlSystem) previamente definido mediante la función cargar\_recomendabilidad().
* Se alimenta el simulador con los valores de **edad**, **interés** y **apertura**.
* El sistema calcula una **puntuación de recomendabilidad** (entre 0 y 100), y según el valor obtenido, se clasifica en tres niveles:
  + baja (≤ 40)
  + media (≤ 70)
  + alta (> 70)

Este nivel no se muestra directamente al usuario, sino que es un **parámetro interno** que luego condiciona el razonamiento del sistema experto.

### 3. Ontología (RDF/RDFS con RDFlib)

Previamente, se ha construido una ontología que define las clases Serie, Pelicula, Contenido, Genero, etc., junto con propiedades como tieneGenero, tieneIdioma, disponibleEn, entre otras. Esta ontología sirve como **estructura semántica** para representar conocimiento.

Durante la carga del motor experto, se generan **hechos** a partir de esta ontología:

* Cada instancia de serie o película se convierte en un **hecho** en el sistema experto.
* Se usa esta información para enriquecer el razonamiento, ya que los hechos tienen propiedades como género, idioma, puntuación, etc.

### 4. Sistema experto (Experta)

Con los hechos de películas y series ya cargados, el sistema experto toma decisiones en función de:

* Las **preferencias del usuario** (Usuario, Preferencia, PlataformaPreferida).
* El **nivel de recomendación difusa** (NivelRecomendacion).
* Las **reglas definidas** en el motor (MotorRecomendacion), que combinan todos estos hechos para inferir la mejor opción posible.

Una vez activado (motor.run()), el motor analiza los hechos y dispara reglas según correspondan, generando una **recomendación final** que incluye:

* El nombre de la película o serie recomendada.
* Su puntuación o justificación interna.

### 5. Presentación de resultados

Finalmente, la interfaz **muestra**:

* El nivel de recomendación calculado por el sistema difuso.
* La recomendación inferida por el sistema experto, si existe.