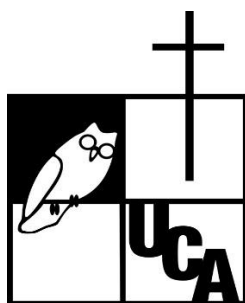


UNIVERSIDAD CENTROAMERICANA JOSE SIMEON CAÑAS

Programación declarativa

Taller 1



Alumno:

Mauricio Alejandro Contreras Montoya

Código:

00000422

San Salvador, 11 octubre 2025

Contenido

1. ¿Qué elementos componen tu base de conocimiento (hechos, relaciones y reglas)?.....	3
2. Escribe una regla en forma de Cláusula de Horn y explica su propósito	4
3. Muestra una consulta con variable libre y otra con variable ligada.	5
4. Representa o describe el árbol de resolución de una de tus consultas.	6
5. Explica el caso trivial y el caso general del ejemplo recursivo que desarrollaste.	6
6. Describe cómo aplicaste los predicados \+, ! y fail y cuál fue su efecto.	7
7. Reflexiona: ¿de qué manera tu base de conocimiento puede considerarse un sistema inteligente lógico?.....	8

Preguntas de Análisis y Reflexión

1. ¿Qué elementos componen tu base de conocimiento (hechos, relaciones y reglas)?

Hechos

- Tipo/Rol: es(a,b).
- Conocimiento: sabe(a,b).
- Amistad: amigo_de(a,b).
- Enseñanza: enseña (a,b)
- Ayuda: ayuda_en(a,b).

Reglas

R1: Sabio	sabio(X):-sabe(X,_),enseña(X,_).	X es sabio si sabe algo y enseña a alguien
R2: Estudioso	estudioso(X):-es(X,aprendiz),sabe(X,_).	X es estudioso si es aprendiz Y sabe algo.
R3: Aliado	aliado(X,Y):-ayuda_en(X,Z),ayuda_en(Y,Z),X\=Y.	X es aliado de si ayudan en la misma tarea (Z) X no es Y .
R4: Conocedor	conocedor(X):-sabe(X,A),sabe(X,B), A\=B.	X es conocedor si sabe dos cosas distintas (a y b).
R5: Maestro	maestro(X):- enseña(X,_),\+ es(X,aprendiz).	X es maestro si enseña a alguien Y no es aprendiz.
Recursividad	maestro_indirecto(X,Y):-enseña(X,Y).	Define una relación de enseñanza

	<pre>maestro_indirecto(X,Y):- enseña(X,Z), maestro_indirecto(Z,Y).</pre>	que puede ser directa (caso base) o encadenada a través de varios aprendices (caso general).
Corte	<pre>consejo_principal(X) :- es(X, mago), !. consejo_principal(X) :- es(X, aprendiz), !. consejo_principal(X) :- es(X, bibliotecaria), !.</pre>	Define una prioridad, donde la respuesta se restringe al primer rol encontrado, gracias al operador de corte (!).
Fallo	<pre>mostrar_personajes :- es(X, _), write('Personaje: '), write(X), nl, fail. mostrar_personajes.</pre>	Usa el predicado fail para forzar la iteración a través de todos los personajes y realizar una acción de escritura.

2. Escribe una regla en forma de Cláusula de Horn y explica su propósito.

sabio(X):-sabe(X,_),enseña(X,_).

Esta regla establece que para que un individuo sea considerado un sabio, deben cumplirse las siguientes dos condiciones en la base de conocimiento:

- El individuo tiene un registro de poseer algún conocimiento o habilidad.
- El individuo tiene un registro de enseñar o transmitir conocimiento a alguien más.

Se infiere que es un sabio, si sabe algo y enseña algo.

3. Muestra una consulta con variable libre y otra con variable ligada.

Variable libre:

enseña(Maestro, Aprendiz).

Buscará todas las combinaciones posibles donde el predicado enseña/2 sea verdadero.

Variable Instanciada:

enseña(merlon, X).

Te dirá a quién enseña Merlón.

4. Representa o describe el árbol de resolución de una de tus consultas.

```
Consulta: sabio(merlon)?  
|  
├─ regla: sabio(X) :- sabe(X,_), enseña(X,_)  
    │      sustituyendo X = merlon  
|  
├─ Subobjetivo 1: sabe(merlon, _)  
    │  
    │   └─ hecho: sabe(merlon, razonamiento_simbolico)  
    │       └─ ✓ satisface el primer subobjetivo  
|  
└─ Subobjetivo 2: enseña(merlon, _)  
    │  
    │   └─ hecho: enseña(merlon, aldara)  
    │       └─ ✓ satisface el segundo subobjetivo
```

Resultado True

5. Explica el caso trivial y el caso general del ejemplo recursivo que desarrollaste.

El caso trivial o caso base:

Es la condición más simple y directa que detiene la recursión. En este caso, define la relación sin necesidad de volver a llamarse a sí misma.

```
maestro indirecto(X,Y):-enseña(X,Y).
```

Establece que X es un maestro indirecto de Y si X le enseña directamente a Y .Se resuelve la consulta sin necesidad de buscar un intermediario, evitando el bucle.

Caso Recursivo:

Define la relación en términos de una versión más pequeña o simple de sí misma. Es el que permite la expansión de la cadena.

maestro_indirecto(X,Y):-

enseña(X,Z),maestro_indirecto(Z,Y).

Establece que X es un maestro indirecto de Y si X le enseña a Z que es un maestro indirecto de Y. La definición se llama a sí mismo. La recursión continua hasta que la ultima llamada encuentra el caso base Z enseña a Y

6. Describe cómo aplicaste los predicados \+, ! y fail y cuál fue su efecto.

Predicado	Regla donde se aplico	Propósito	Efecto
\+	R5 Maestro	Definir a un maestro como alguien que enseña Y no es un aprendiz.	El predicado \+ es(X,aprendiz) solo tiene éxito si la meta interna es(X,aprendiz) falla. Esto asegura que, aunque Merlon enseña, no será considerado maestro si su rol fuera el de aprendiz.

i	consejo_principal(X)	Establecer una prioridad estricta para la elección del consejero, basándose en el orden de las reglas.	Una vez que se encuentra la primera solución (es(X, mago)) y se ejecuta el corte, Prolog descarta irreversiblemente las opciones subsiguientes (es(X, aprendiz) y es(X, bibliotecaria)), sin permitir retroceso a ellas.
fail	mostrar_personajes	Obligar a Prolog a retroceder y encontrar todas las soluciones para es(X,_) de forma secuencial, imprimiendo cada una.	El fail en el cuerpo de la primera cláusula fuerza al motor de inferencia a reintentar la meta anterior, es(X,_), una y otra vez hasta agotar todos los personajes, logrando un ciclo de impresión.

7. Reflexiona: ¿de qué manera tu base de conocimiento puede considerarse un sistema inteligente lógico?

Se puede considerar un sistema inteligente lógico porque utiliza la lógica de primer orden a través de cláusulas de horn para realizar inferencias y retornar nuevos conocimientos que no están explícitos declarados.

La base encapsula conocimiento de dos maneras esenciales.

- Hechos: Conocimiento explícito y seguro.

- Reglas: Conocimiento inferencial y abstracto. Permite que el sistema razone un poco más allá de los datos brutos

Además, puede llegar a conclusiones validas a partir de ellos por medio de deducciones o búsqueda(backtracking).Es capaz de utilizar el conocimiento para responder a consultas que son esencialmente problemas que se presentan de verificación o generación de soluciones

```
1 ?- ensena(Maestro,Aprendiz).  
Maestro = merlon,  
Aprendiz = aldara.
```

```
2 ?- ensena(merlon,X).  
X = aldara.
```

```
2 ?- aliado(X,Y).  
X = gorik,  
Y = lyra .
```

```
3 ?- ensena(merlon,X),sabe(X,_).  
X = aldara.
```

```
4 ?- sabio(X),ayuda_en(X,Tarea).  
X = merlon,  
Tarea = leyes_basicas ;  
X = merlon,  
Tarea = leyes_basicas ;  
false.
```

```
5 ?- maestro_indirecto(merlon,aldara).  
true
```

```
6 ?- maestro_indirecto(merlon,aprendiz2).  
false.
```

```
7 ?- maestro_indirecto(aldara,aprendiz2).  
false.
```

```
8 ?- consejo_principal(X).  
X = merlon.
```

```
9 ?- mostrar_personajes  
.   
Personaje: merlon  
Personaje: aldara  
Personaje: gorik  
Personaje: lyra  
true.
```