

# La lógica de Avengers. (Diciembre, 2023)

Daza Pereira Juan Pablo y Camargo Sanchez Juan Sebastian

Fundamentos de la Inteligencia Artificial

Escuela Colombiana de Ingeniería Julio Garavito

**Resumen-** En este proyecto implementaremos la lógica argumental de la famosa película de Marvel Avengers:Endgame, tomando el lore del universo Cinematográfica hasta el final de película. El problema surge cuando se tiene un universo tan grande y con tantos hechos, que se puede dificultarse seguir las líneas argumentales de las películas. Por lo que nos tomaremos el trabajo de plasmar estos hechos por medio de la lógica predicados usando como motor de inferencia Prolog , la cual nos facilitara el seguimiento de las líneas argumentales y las relaciones entre los personajes y eventos. Dado que estamos tomando hechos que se pueden comprobar y que nosotros mismo como espectadores frecuentes de este contenido, nos consideramos expertos en el tema.

## INTRODUCCIÓN.

La película de Avengers:Endgame, el punto central de este proyecto es la segunda parte de la saga del infinito de Marvel. La primera parte de la saga vemos a varios héroes reunidos para derrotar una gran amenaza que planea acabar con el 50% de la vida del universo. Thanos, el villano de este plan tiene éxito y elimina a la mitad del universo con un “chasquido”, en esto varios héroes también mueren y es aquí donde las cosas se empiezan a complicar. Son varias muertes y como espectador no sabes muy bien lo que paso, hasta que veas la segunda película. Ya en la segunda película se resuelve el conflicto, con viajes en el tiempo y multiversos, algunos personajes que murieron reviven y otros se quedan muertos. La línea argumental se vuelve complicada por lo que nos plantemos hacer un análisis de los eventos y de los personajes. Su estado, ubicación, ¿asesino alguien?, ¿fue asesinado? Y hacer una abstracción de este conocimiento a la lógica de predicados. Con esto le podremos consultar sobre algún evento, como puede ser ¿quién asesino a este héroe? ¿Qué ubicación se encuentra? Entre otros. Permitiendo un análisis detallado y consultas específicas sobre eventos y personajes en el universo cinematográfico de Marvel.

## MARCO TEÓRICO.

La lógica de predicados y Prolog desempeñan un papel fundamental en la representación del conocimiento y la inferencia en sistemas de inteligencia artificial. En este contexto, vamos a explorar cómo estos conceptos se relacionan y pueden aplicarse en la implementación de la lógica argumental de Avengers: Endgame.

Lógica de Predicados en Inteligencia Artificial:

La lógica de predicados es un formalismo lógico que permite representar relaciones y propiedades entre objetos mediante predicados y cuantificadores. En el ámbito de la inteligencia artificial (IA), la lógica de predicados se utiliza para modelar el conocimiento de manera declarativa, lo que facilita la representación y manipulación de hechos complejos.

Predicados y Relaciones:

Los predicados representan propiedades o relaciones entre objetos. Por ejemplo, en Avengers: Endgame, podríamos tener predicados como "Heroe(Personaje)", "ubicacion(Personaje, Lugar)", etc.

Cuantificadores:

Los cuantificadores, como "para todo" y "existencial", permiten expresar afirmaciones generales o particulares. Esto puede ser útil para abordar eventos que involucren a múltiples personajes.

Inferencia Lógica:

La inferencia lógica en la lógica de predicados implica deducir nueva información a partir de reglas y hechos existentes. Esto es esencial para analizar y entender las relaciones complejas entre los eventos en el universo de Avengers.

Backtracking:

Comienza la búsqueda de soluciones aplicando reglas de inferencia y sustituyendo variables para intentar satisfacer el objetivo y las restricciones (OR/AND).

Si se encuentra en un punto donde las reglas de inferencia no pueden continuar o se violan las restricciones, retrocedes (backtrack) al estado anterior y explora otras posibilidades.

Prolog:

Prolog es un lenguaje de programación lógica que se basa en la lógica de predicados. Su sintaxis simple y su capacidad para expresar relaciones lógicas lo hacen idóneo para modelar y consultar conocimientos complejos de manera eficiente.

Hechos y Reglas en Prolog:

En Prolog, los hechos son afirmaciones sobre el mundo, y las reglas establecen relaciones lógicas. En el contexto de Avengers: Endgame, podríamos tener reglas como "ha\_sido\_asesinado\_por\_heroe(Personaje)" o "(heroes\_asesinados\_por\_villano(Villano, HeroesAsesinados) ".

Consultas:

Prolog permite realizar consultas para obtener información específica. Por ejemplo, podríamos consultar "¿quién asesinó a este héroe?" o "¿dónde se encuentra este personaje?".

Manejo de Excepciones:

Prolog permite manejar excepciones lógicas y desconocimiento, lo cual es útil para abordar eventos ambiguos o desconocidos en el universo de Avengers.

## **METODOLOGÍA.**

Para montar este sistema experto tomaremos como base de hechos lo básico y general para cualquier espectador, que son quienes son los héroes, villanos, su ubicación, su estado (vivo o muerto), objeto y si es parte del equipo de vengadores. Después de ya como expertos le vamos a incluir varios predicados, con los cuales podremos hacer consultas sobre los eventos y hechos que pasan en la película, pero sin ser tan generales como quien es héroe o quien es villano.

En la definición de Héroes y Villanos utilizaremos la función `heroe(X)` y `villano(X)`, siendo X cualquier personaje de universo Marvel.

Ahora para las reglas definiremos algunas reglas que creemos que son útiles y que se puede deducir a partir de la lógica de predicados para la consulta de cualquier espectador. Estas serán:

```
% Reglas
% Regla para identificar si dos personajes están en el mismo equipo.
%Consulta: en_mismo_equipo(*Personaje1, *Personaje2).
en_mismo_equipo(Personaje1, Personaje2) :-
    (vengador(Personaje1); aliados(Personaje1, _)),
    (vengador(Personaje2); aliados(Personaje2, _)),
    Personaje1 \= Personaje2,
    (
        vengador(Personaje1), vengador(Personaje2);
        aliados(Personaje1, Equipo), aliados(Personaje2, Equipo)
    ).
```

```
% Regla para determinar si un héroe posee un objeto.
%Consulta: posee_objeto(*Heroe, Objeto).
posee_objeto(Heroe, Objeto) :- objeto(Heroe, Objeto).
```

```
%Regla para encontrar todos los aliados de un personaje.
%Consulta: todos_los_aliados_de(*Personaje, Aliados).
todos_los_aliados_de(Personaje, Aliados) :-
    findall(Aliado, aliados(Personaje, Aliado), Aliados).
```

```
% Regla para determinar si un personaje ha sido asesinado por un villano.
%Consulta: asesinado_por_villano(*Personaje).
asesinado_por_villano(Personaje) :-
    heroe(Personaje),
    mato(Villano, Personaje),
    villano(Villano).
```

```
%Regla para determinar si un personaje ha sido asesinado por un héroe.
%Consulta: ha_sido_asesinado_por_heroe(*Personaje).
ha_sido_asesinado_por_heroe(Personaje) :-
    heroe(Heroe), villano(Personaje), mato(Heroe, Personaje).
```

```
%Regla para encontrar todos los héroes que han sido asesinados por un villano en particular.
%Consulta: heroes_asesinados_por_villano(*Villano, HeroesAsesinados).
heroes_asesinados_por_villano(Villano, HeroesAsesinados) :-
    villano(Villano),
    findall(Heroe, (mato(Villano, Heroe), heroe(Heroe)), HeroesAsesinados).
```

```
%Regla para encontrar si un héroe es vengador heroes_en_ubicacion.
%Consulta: es_vengador(*Heroe).
es_vengador(Heroe) :- vengador(Heroe).
```

```
%Regla para mostrar los héroes que estuvieron en cierta ubicación.
%Consulta: heroes_en_ubicacion(Heroe, *Ubicacion).
heroes_en_ubicacion(Heroe, Ubicacion) :- estuvo_en(Heroe, Ubicacion), heroe(Heroe).
```

```
%Reglas que busca determinar si un héroe es un Vengador y además estuvo en una ubicación específica.
%Consulta: vengador_en_ubicacion(*Heroe, *Ubicacion)
vengador_en_ubicacion(Heroe, Ubicacion) :-
    vengador(Heroe),
    estuvo_en(Heroe, Ubicacion).
```

Como motor de inferencia usaremos Prolog ya que este con este lenguaje además de ser para lógica de predicados nos permite usar su algoritmo de backtracking que ya tiene implementado.

## RESULTADOS

Como se mencionó anteriormente, vamos a tomar para el desarrollo de la lógica de los vengadores como punto central el final de la película de Avengers End Game y todo anterior desarrollo (desde *Avengers 1* a *Avengers End Game*) siguiendo la línea secuencial de la saga, pero centrándonos solamente en la última película y en punto final de esta, suponiéndose que ya se conoce todo lo sucedido en la película.

Empezamos dividiendo la idea en tres etapas. Una etapa inicial que es la clasificación de la base de hechos, una etapa central en la que se desarrollan las reglas de la lógica y, por último, una etapa final en donde se escriben las consultas y se desarrollan las pruebas con base en el conocimiento de los expertos (nosotros los creadores del proyecto).

Para la etapa inicial, que es la creación y clasificación de la base de hechos, quisimos centrarnos en los personajes, objetos y sucesos más importantes de la película que dieron un gran impacto a los amantes de la saga y sus espectadores. En la película se presentaron más de 30 personajes los cuales tomamos los personajes que están desde el inicio de la saga y los personajes que tuvieron un gran impacto en la película ya fuera haciendo una gran pelea, derrotando a otro personaje o realizando una gran hazaña. Después de tener esos personajes tan importantes, los dividimos en dos categorías: Héroes y Villanos, los cuales son:

- **Héroes:**

- Capitán América.



- Iron Man.



➤ Thor.



➤ Hulk.



➤ Viuda negra.



➤ Ojo de halcón.



➤ Máquina de guerra



➤ Spider-Man.



➤ La bruja escarlata.



➤ Halcón.



➤ Capitana Marvel.



➤ Star lord.



➤ Doctor Strange.



➤ Ant Man.



➤ El soldado del invierno.



➤ Loki.





➤ Nébula.



➤ Rocket



➤ Groot.



➤ Gamora.



➤ Drax.



➤ Black panther



➤ Okoye.



- ***Villanos:***

➤ Thanos.



➤ Ebony maw.



➤ Proxima midnight.



➤ Corvus glaive.



➤ Cull obsidian.



➤ Ejército de Thanos.



Planteando así la base de hechos:

- *heroe(X)*.
- *villano(X)*.

Luego de tomar los personajes, nos centramos en los objetos que estos usaron, aclarando que también tomamos los objetos que tuvieron relevancia en la película junto con el personaje que lo portaba, teniendo los siguientes resultados:

- **Objetos:**
  - Thor portaba el mjolnir.
  - Thor portaba el rompe tormentas.
  - Capitán América portaba el escudo.
  - Capitán América portaba el mjolnir.
  - Iron Man portaba la armadura.
  - Viuda negra portaba las pulseras.
  - Ojo de halcón portaba el arco.
  - Máquina de guerra portaba la armadura.
  - Halcón portaba la armadura.
  - Star lord portaba un arma.
  - Doctor Strange portaba el sling ring.
  - El soldado del invierno portaba un brazo de acero.
  - Loki portaba un cuchillo.
  - Rocket portaba un arma.
  - Black panther portaba una armadura.
  - Okoye portaba una lanza.
  - Thanos portaba el guantelete del infinito.

Planteando así la base de hechos:

- *objeto(X, Y)*, siendo X el héroe/villano y Y el objeto.

Por consiguiente, se tuvo la idea de ver los personajes que murieron en la película y el personaje que los mató, porque algunas de esas muertes causaron alegría a los espectadores, como otras que nos rompieron en pedazos. Dejando así las muertes que sucedieron en la película:

- **Mató:**

- Thanos mató a Iron Man.
- Iron Man mató a Thanos.
- Ojo de halcón mató a la Viuda negra.
- Thanos mató a Loki.
- Spider-Man mató a Ebony Maw.
- Iron Man mató a Ebony Maw.
- Hulk mató a Cull Obsidian.
- Okoye mató a Corvus Glaive.
- Iron Man mató a Proxima Midnight.
- Iron Man mató al Ejército de Thanos.

Quedando la base de hechos de la siguiente manera:

- *mato(X, Y)*, donde X e Y son personajes de la película.

Siguiendo con la idea, quisimos poner el equipo original de las películas de los Avengers y ver si todos ellos seguían aún presentes en la última entrega, mostrando los siguientes resultados:

- **Vengadores:**
  - Capitán América.
  - Hulk.
  - Iron Man.
  - Thor.
  - Viuda negra.
  - Ojo de halcón.
  - Máquina de guerra.
  - Spider-Man.
  - Bruja Escarlata.
  - Halcón.

Teniendo la siguiente base de hechos:

- *vengador(X)*, donde X es un héroe.

Luego quisimos ver el tipo de alianzas que hicieron los vengadores en la película para derrotar a Thanos, a sus hijos y a su ejército, teniendo las siguientes alianzas:

- **Aliados:**

- Capitana Marvel se alió con los vengadores.
- Ant man se alió con los vengadores.
- Black panther se alió con los vengadores.
- Okoye se alió con los vengadores.
- Star lord se alió con los vengadores.
- Loki se alió con los vengadores.
- Doctor Strange se alió con los vengadores.
- Nebula se alió con los vengadores.
- Rocket se alió con los vengadores.
- Groot se alió con los vengadores.
- Gamora se alió con los vengadores.
- Drax se alió con los vengadores.
- Capitana Marvel se alió con SHIELD.
- Los vengadores se aliaron con los Guardianes de la Galaxia.
- Los vengadores se aliaron con el ejército de Wakanda.
- Los vengadores se aliaron con SHIELD.
- Los vengadores se aliaron con los Maestros de las artes místicas.

Quedando la base de hechos de la siguiente forma:

- *aliados(X, Y)*, donde X es un persona e Y es un aliados.

Como penúltima parte, introducimos las ubicaciones más importantes de la película donde se realizaron las batallas por salvar la existencia del universo, eligiendo los siguiente:

- **Ubicación:**

- New York.
- Asgard (lugar de origen de Thor y Loki).
- Vormir (planeta de la gema del alma).
- Reino Cuántico.
- Avengers headquarters (base de los vengadores).
- Planeta Titan (lugar de origen de Thanos).
- Tokio.
- Morag (planeta de la gema del poder).

Quedando la base de hechos de la siguiente manera:

- *ubicación(X)*, donde X es una ubicación.

Y ya por último, quisimos mostrar los personajes que estuvieron en esas ubicaciones, definiendo la base de hechos de la siguiente manera:

- *estuvo\_en(X, Y)*, donde X es un personaje e Y es una ubicación.

Luego de tener definida y planteada la Fase 1 de desarrollo, pasamos a las siguientes fases, la Fase 2 donde se desarrollaron las reglas que nos iban a permitir abarcar toda la base de hechos y poder crear reglas que nos dieran un contexto general de la película, y la Fase 3 donde se realizaron las consultas y se realizaron las pruebas corroboradas por los expertos (nosotros los creadores del proyecto). Las reglas se mostraron anteriormente en la metodología, pero aquí les haremos un análisis profundo para dejar bien claro las funciones que cumplen cada una.

- **Regla #1: Regla para identificar si dos personajes están en el mismo equipo.**

```
en_mismo_equipo(Personaje1, Personaje2) :-  
    (vengador(Personaje1); aliados(Personaje1, _)),  
    (vengador(Personaje2); aliados(Personaje2, _)),  
    Personaje1 \= Personaje2,  
    (  
        vengador(Personaje1), vengador(Personaje2);  
        aliados(Personaje1, Equipo), aliados(Personaje2, Equipo)  
    ).
```

La función que cumple esta regla es la de dar a conocer los héroes que comparten mismo equipo. Como se ve en la regla, se pide insertar dos personajes de la película, y la misma regla es la que se encarga de leer toda la base de hechos para encontrar los personajes que comparten equipo por medio de los vengadores y de las alianzas que hay.

Para comprobar esta regla, se realizaron seis pruebas, donde se espera que el 50% de ellas sean verdaderas y otro restante no lo sea. A continuación mostramos los resultados:

- ✓ Aprobadas:

```
en_mismo_equipo(thor, hulk)
true
Next 10 100 1,000 Stop
?- en_mismo_equipo(thor, hulk)
```

```
en_mismo_equipo(star_lord, groot)
true
?- en_mismo_equipo(star_lord, groot)
```

```
en_mismo_equipo(iron_man, viuda_negra)
true
Next 10 100 1,000 Stop
?- en_mismo_equipo(iron_man, viuda_negra)
```

✓ Rechazadas:

```
en_mismo_equipo(star_lord, viuda_negra)
false
?- en_mismo_equipo(star_lord, viuda_negra)
```

```
en_mismo_equipo(iron_man, nebula)
false
?- en_mismo_equipo(iron_man, nebula)
```

```
en_mismo_equipo(drax, viuda_negra)
false
?- en_mismo_equipo(drax, viuda_negra)
```

- **Regla #2: Regla para determinar si un héroe posee un objeto.**

```
posee_objeto(Heroe, Objeto) :- objeto(Heroe, Objeto).
```

Esta regla es simple, es para dar a conocer el tipo de objeto que usa un héroe o villano. Para esta regla también se realizaron seis pruebas, y los resultados fueron los siguientes:



✓ Aprobadas:

```
posee_objeto(iron_man, Objeto)
Objeto = armadura
Objeto = guantelete_del_infinito
?- posee_objeto(iron_man, Objeto)
```

```
posee_objeto(thor, Objeto)
Objeto = mjolnir
Objeto = rompe_tormentas
?- posee_objeto(thor, Objeto)
```

```
posee_objeto(capitan_america, Objeto)
Objeto = escudo
Objeto = mjolnir
?- posee_objeto(capitan_america, Objeto)
```

✓ Rechazadas:

```
posee_objeto(hulk, Objeto)
false
?- posee_objeto(hulk, Objeto)
```

```
posee_objeto(capitana_marvel, Objeto)
false
?- posee_objeto(capitana_marvel, Objeto)
```

```
posee_objeto(spiderman, Objeto)
false
?- posee_objeto(spiderman, Objeto)
```

Las pruebas que lanzan de resultado false, es porque esos héroes o villanos no poseen algún objeto.

- **Regla #3: Regla para encontrar todos los aliados de un personaje.**

```
todos_los_aliados_de(Personaje, Aliados) :-  
    findall(Aliado, aliados(Personaje, Aliado), Aliados).
```

Esta regla lo que hace es mostrar los aliados que tiene un héroe mostrándolos en una lista seguida. A continuación se muestran las seis pruebas realizadas:

✓ Aprobadas:

```
todos_los_aliados_de(star_lord, Aliados)  
Aliados = [vengadores]  
?- todos_los_aliados_de(star_lord, Aliados)
```

```
todos_los_aliados_de(vengadores, Aliados)  
Aliados = [guardianes_de_la_galaxia, ejercito_de_wakanda, shield, maestros_de_las_artes_misticas]  
?- todos_los_aliados_de(vengadores, Aliados)
```

```
todos_los_aliados_de(dr_strange, Aliados)  
Aliados = [vengadores]  
?- todos_los_aliados_de(dr_strange, Aliados)
```

```
todos_los_aliados_de(thanos, Aliados)  
Aliados = []  
?- todos_los_aliados_de(thanos, Aliados)
```

```
todos_los_aliados_de(iron_man, Aliados)  
Aliados = []  
?- todos_los_aliados_de(iron_man, Aliados)
```

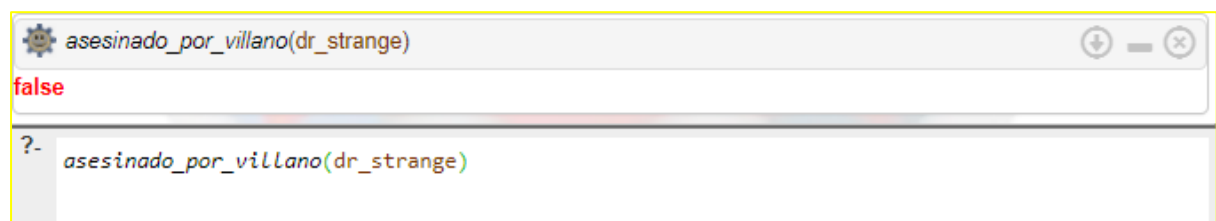
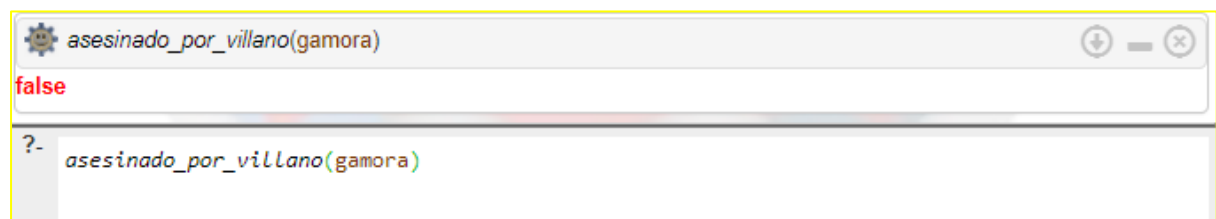
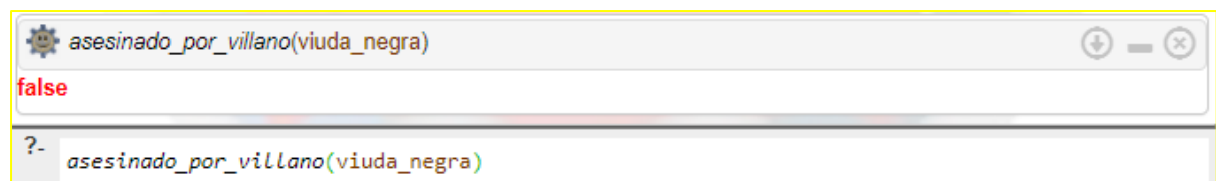
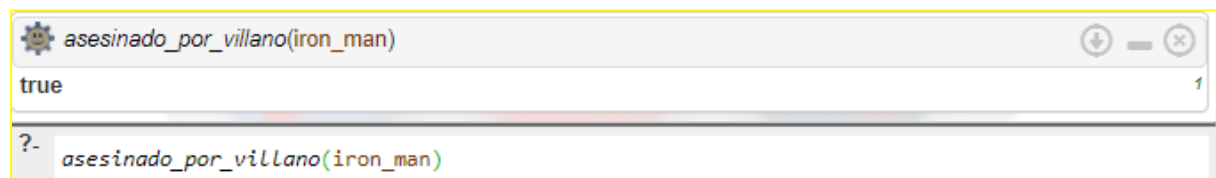
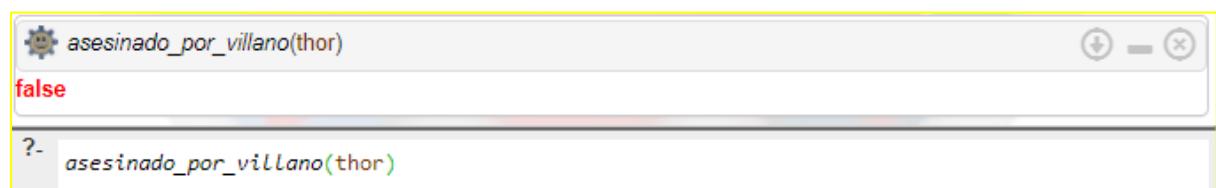
```
todos_los_aliados_de(thor, Aliados)  
Aliados = []  
?- todos_los_aliados_de(thor, Aliados)
```

Todas las pruebas fueron aprobadas, sea el tipo de héroe o villano que se consulte, siempre son correctas. Las listas que salen vacías están puesto que no todos los héroes o villanos no tienen aliados.

- **Reglas #4: Regla para determinar si un personaje ha sido asesinado por un villano.**

```
asesinado_por_villano(Personaje) :-  
    heroe(Personaje),  
    mato(Villano, Personaje),  
    villano(Villano).
```

Esta regla determina si algún héroe de la saga ha sido asesinado, dando a conocer por medio de la consulta, si el héroe fue asesinado o no, probándolo de la siguiente manera:



- **Regla #5: Regla para determinar si un personaje ha sido asesinado por un héroe.**

```
ha_sido_asesinado_por_heroe(Personaje) :-
    heroe(Heroe), villano(Personaje), mato(Heroe, Personaje).
```

Al igual que la regla, se quiere determinar si un villano fue o no asesinado por un héroe en la película, probándolo de la siguiente manera:

```
ha_sido_asesinado_por_heroe(thanos)
true
?- ha_sido_asesinado_por_heroe(thanos)
```

```
ha_sido_asesinado_por_heroe(ejercito_de_thanos)
true
?- ha_sido_asesinado_por_heroe(ejercito_de_thanos)
```

```
ha_sido_asesinado_por_heroe(ebony_maw)
true
?- ha_sido_asesinado_por_heroe(ebony_maw)
```

- **Regla #6: Regla para encontrar todos los héroes que han sido asesinados por un villano en particular.**

```
heroes_asesinados_por_villano(Villano, HeroesAsesinados) :-
    villano(Villano),
    findall(Heroe, (mato(Villano, Heroe), heroe(Heroe)), HeroesAsesinados).
```

Las reglas 4 y 5 solamente mostraban una afirmación de si el héroe o villano ha sido asesinado, pero esta quiere mostrar el nombre de los héroes asesinados y su asesino, probándolo de la siguiente manera:

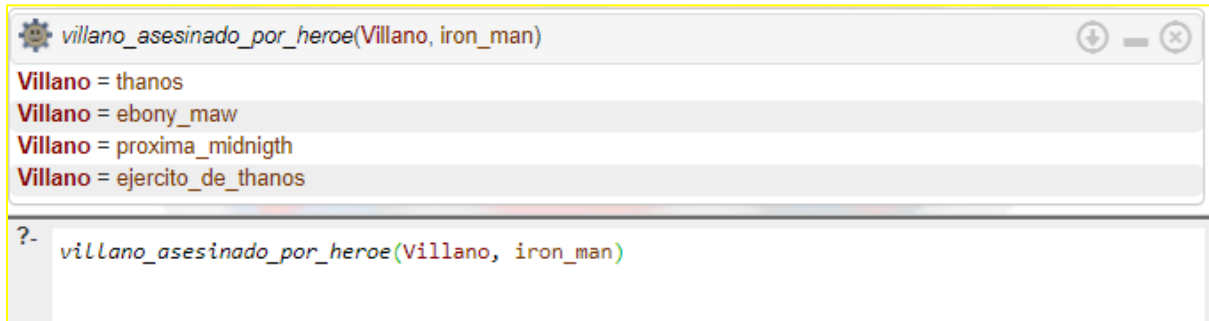
```
heroes_asesinados_por_villano(thanos, HeroesAsesinados)
HeroesAsesinados = [iron_man, loki]
?- heroes_asesinados_por_villano(thanos, HeroesAsesinados)
```

Esta es la única consulta que se puede realizar, puesto que, en el contexto de la película, el único villano que asesinó a los héroes fue Thanos. Si se realizan otra consulta con los demás villanos, estas serán falsas.

- **Regla #7: Regla para encontrar todos los villanos que han sido asesinados por un héroe en particular.**

```
villano_asesinado_por_heroe(Villano, Heroe) :-
    heroe(Heroe),
    villano(Villano),
    mato(Heroe, Villano).
```

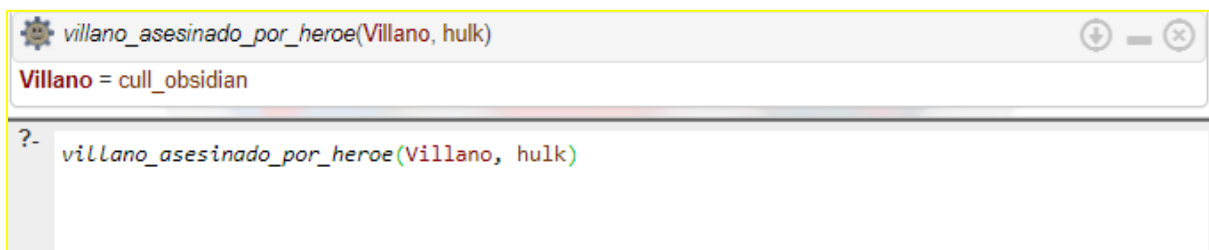
Al igual que en la regla anterior, aquí se quiere mostrar el nombre de los villanos asesinados y su asesino, corroborado de la siguiente manera:



```
villano_asesinado_por_heroe(Villano, iron_man)
```

Villano = thanos  
Villano = ebony\_maw  
Villano = proxima\_midnigth  
Villano = ejercito\_de\_thanos

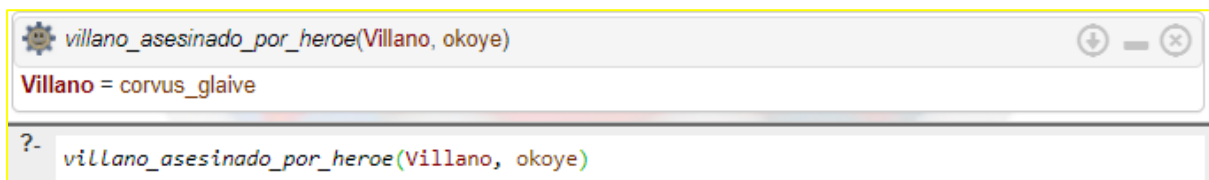
```
?- villano_asesinado_por_heroe(Villano, iron_man)
```



```
villano_asesinado_por_heroe(Villano, hulk)
```

Villano = cull\_obsidian

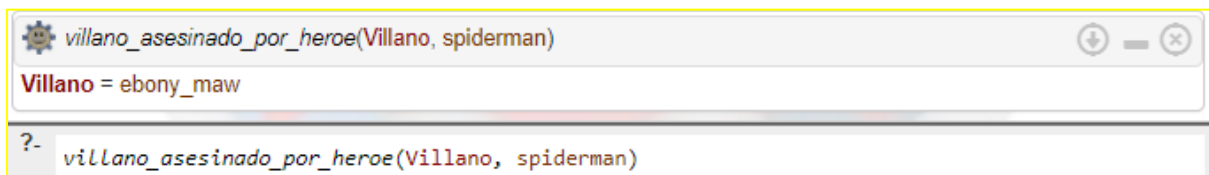
```
?- villano_asesinado_por_heroe(Villano, hulk)
```



```
villano_asesinado_por_heroe(Villano, okoye)
```

Villano = corvus\_glaive

```
?- villano_asesinado_por_heroe(Villano, okoye)
```



```
villano_asesinado_por_heroe(Villano, spiderman)
```

Villano = ebony\_maw

```
?- villano_asesinado_por_heroe(Villano, spiderman)
```

- **Regla #8: Regla para encontrar si un héroe es vengador.**

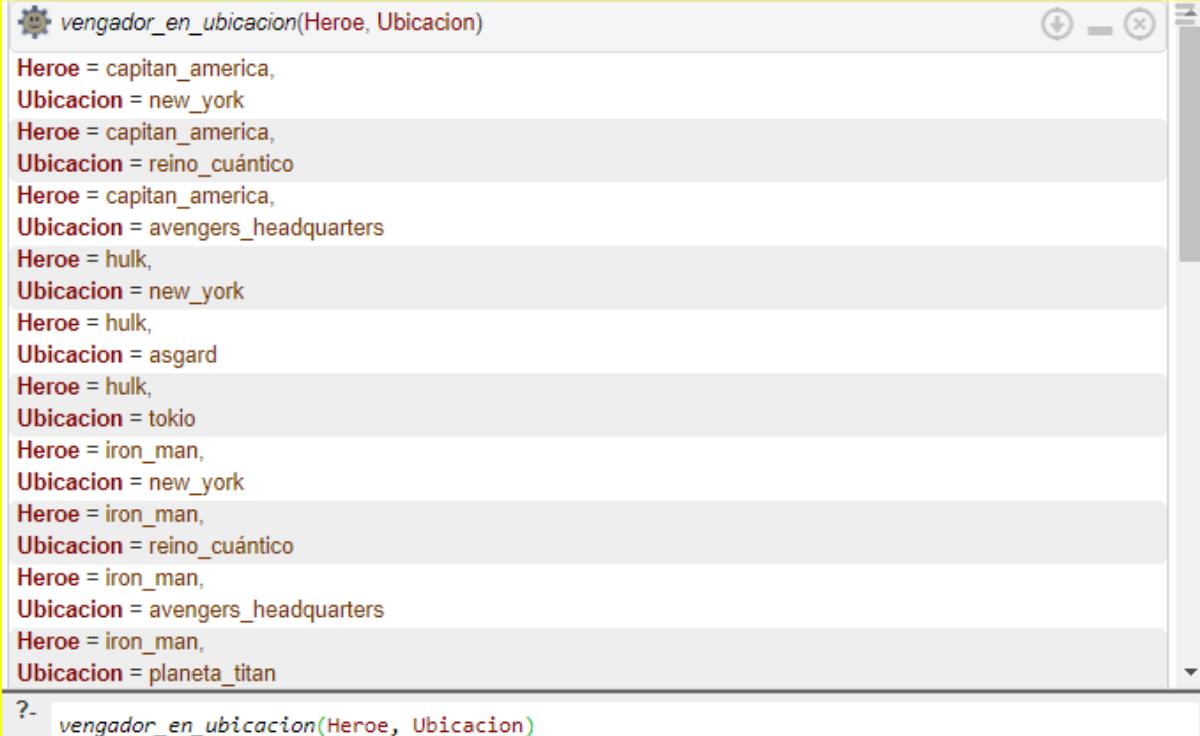
```
es_vengador(Heroe) :- vengador(Heroe).
```

Esta regla es simple, y quiere mostrar si los vengadores originales siguen en la última película y corroborar que si se inserta otro personaje es un vengador o no.

- **Regla #9: Regla para mostrar los héroes que estuvieron en cierta ubicación.**

```
heroes_en_ubicacion(Heroe, Ubicacion) :- estuvo_en(Heroe, Ubicacion), heroe(Heroe).
```

Esta regla es muy general y permite dar a conocer la mayoría del contexto de la película puesto que muestra las ubicaciones donde se situaron los vengadores:



The screenshot shows a Prolog IDE window titled 'vengador\_en\_ubicacion(Heroe, Ubicacion)'. The main area displays a list of 15 results, each consisting of a hero name and a location. The results are: capitan\_america, new\_york; capitan\_america, reino\_cuántico; capitan\_america, avengers\_headquarters; hulk, new\_york; hulk, asgard; hulk, tokio; iron\_man, new\_york; iron\_man, reino\_cuántico; iron\_man, avengers\_headquarters; and iron\_man, planeta\_titan. The results are displayed in alternating light and dark gray rows. At the bottom, there is a prompt '?-' followed by the query 'vengador\_en\_ubicacion(Heroe, Ubicacion)'.

```
vengador_en_ubicacion(Heroe, Ubicacion)  
Heroe = capitan_america,  
Ubicacion = new_york  
Heroe = capitan_america,  
Ubicacion = reino_cuántico  
Heroe = capitan_america,  
Ubicacion = avengers_headquarters  
Heroe = hulk,  
Ubicacion = new_york  
Heroe = hulk,  
Ubicacion = asgard  
Heroe = hulk,  
Ubicacion = tokio  
Heroe = iron_man,  
Ubicacion = new_york  
Heroe = iron_man,  
Ubicacion = reino_cuántico  
Heroe = iron_man,  
Ubicacion = avengers_headquarters  
Heroe = iron_man,  
Ubicacion = planeta_titan  
?- vengador_en_ubicacion(Heroe, Ubicacion)
```

```
Heroe = thor,  
Ubicacion = new_york  
Heroe = thor,  
Ubicacion = asgard  
Heroe = thor,  
Ubicacion = reino_cuántico  
Heroe = thor,  
Ubicacion = avengers_headquarters  
Heroe = thor,  
Ubicacion = avengers_headquarters  
Heroe = viuda_negra,  
Ubicacion = new_york  
Heroe = viuda_negra,  
Ubicacion = reino_cuántico  
Heroe = viuda_negra,  
Ubicacion = avengers_headquarters  
Heroe = viuda_negra,  
Ubicacion = morag  
Heroe = ojo_de_halcon,  
Ubicacion = reino_cuántico  
Heroe = ojo_de_halcon,  
Ubicacion = avengers_headquarters  
?- vengador_en_ubicacion(Heroe, Ubicacion)
```

```
Heroe = spiderman,  
Ubicacion = planeta_titan  
Heroe = spiderman,  
Ubicacion = avengers_headquarters  
Heroe = bruja_escarlata,  
Ubicacion = avengers_headquarters  
Heroe = halcon,  
Ubicacion = avengers_headquarters  
Heroe = halcon,  
Ubicacion = avengers_headquarters  
?- vengador_en_ubicacion(Heroe, Ubicacion)
```

## CONCLUSIONES

- Se puede concluir que por medio de un proyecto como este se pueden abarcar diferentes temas y contextos, permitiendo simplificarlos en unas simples oraciones que abarcan todo lo que se quiere dar a conocer.
- En nuestro contexto, el de la película de Avengers End Game, se pudo cumplir con lo se quería, la idea de poder tomar una película de más de 3 horas de duración en diferentes contextos de espacios y tiempos, pudiendo darle una

herramienta a cualquier persona que no posea conocimientos de la película para que sepa y aprenda de ella.