

1.

	átomos	variables	Ninguno
1. Vincent		X	
2. Masaje de pies			X
3. variable23	X		
4. Variable2000		X	
5. big_kahuna_burger	X		
6. 'Gran hamburguesa kahuna'	X		
7. Hamburguesa Kahuna Grande			X
8. 'Jules'	X		
9. _Jules		X	
10. '_Jules'	X		

2.

	átomos	variables	terminos complejos	Ninguno	funtor	aridad
1. amores (Vincent,mia)				X		
2. 'amores(Vincent,mia)'	X					
3. Butch (boxeador)				X		
4. boxeador (Butch)				X		
5. y (grande (hamburguesa), kahuna (hamburguesa))				X		
6. y(grande(X),kahuna(X))			X		Y	2
7. _and(grande(X),kahuna(X))				X		
8. (Butch mata a Vincent)				X		
9. mata(Butch Vincent)				X		
10. mata(Butch,Vincent)				X		

- 1- Hay un espacio entre el funtor y los argumentos, por lo que no representa nada.
- 2- Es un átomo debido a que se encuentra entre comillas simples.
- 3-4-5- Como están dejando un espacio, no pueden ser nada.
- 6- Es un término complejo, donde "y" es el funtor y la aridad es 2 debido a que tenemos dos argumentos (grande y kahuna).
- 7- En este caso, empieza con un guion bajo, lo cual no es posible, ya que los funtores deben empezar con minúsculas.
- 8- Aquí, no hay funtor, además de que lo que está dentro del paréntesis tiene espacios.
- 9- Hay un espacio entre Butch y Vincent en vez de una coma, por lo que no sería nada.
- 10- En este caso, no se cierra el paréntesis.

3.

woman(vincent).
woman(mia).
man(jules).
person(X):- man(X); woman(X).
loves(X,Y):- father(X,Y).
father(Y,Z):- man(Y), son(Z,Y).
father(Y,Z):- man(Y), daughter(Z,Y).

Los **hechos** son declaraciones simples y directas consideradas verdaderas, sin necesidad de justificación. Por lo tanto, podemos decir que tenemos tres hechos:

- woman(vincent).
- woman(mia).
- man(jules).

En cuanto a las **reglas**, son las relaciones lógicas entre predicados, donde la verdad del predicado de la izquierda (consecuente) depende de la verdad del predicado de la derecha (antecedente). Por lo que podemos encontrar cuatro reglas:

- person(X) :- man(X); woman(X).
- loves(X,Y) :- father(X,Y).
- father(Y,Z) :- man(Y), son(Z,Y).
- father(Y,Z) :- man(Y), daughter(Z,Y).

Las **cláusulas** son el conjunto de reglas y hechos. Por lo tanto, tenemos siete cláusulas (tres hechos y cuatro reglas):

- woman(vincent).
- woman(mia).
- man(jules).
- person(X) :- man(X); woman(X).
- loves(X,Y) :- father(X,Y).
- father(Y,Z) :- man(Y), son(Z,Y).
- father(Y,Z) :- man(Y), daughter(Z,Y).

Los **predicados** son funciones que devuelven un valor de verdad en función de los argumentos que se les proporcionen. En este caso tenemos siete

predicados:

- woman
- man
- person
- loves
- father
- son
- daughter

Cuando hablamos del **encabezado de las reglas**, hacemos referencia a la parte izquierda de una regla que establece lo que se está tratando de comprobar. Entonces, en este caso tendríamos tres encabezados:

- person(X).
- loves(X,Y).
- father(Y,Z).

Y si hablamos de los **objetivos de las reglas**, estamos hablando de la parte derecha de una regla donde se definen las condiciones que deben cumplirse para que el encabezado sea verdadero. Aquí podemos encontrar los siguientes objetivos de las reglas.

- man(X) y woman(X) para la regla: `person(X) :- man(X); woman(X).`
- father(X,Y) para la regla: `loves(X,Y) :- father(X,Y).`
- man(Y) y son(Z,Y) para la regla: `father(Y,Z) :- man(Y), son(Z,Y).`
- man(Y) y daughter(Z,Y) para la regla:  
`father(Y,Z) :- man(Y), daughter(Z,Y).`

4.

- |   |
|---|
| 1. Butch es un asesino.   |
| 2. Mia y Marsellus están casados.                                     |
| 3. Zed ha muerto.   |
| 4. Marsellus mata a todos los que le dan a Mia un masaje en los pies. |
| 5. Mia ama a todos los que son buenos bailarines.                     |
| 6. Jules come cualquier cosa que sea nutritiva o sabrosa.             |

asesino(butch).

casados(mia, marsellus).

casados(marsellus, mia).

muerto(zed).

mata(marsellus, X) :- masajePies(X, mia).

ama(mia, X) :- buenBailarin(X).

come(jules, X) :- sabroso(X). }  
come(jules, X) :- nutritivo(X). } come(jules, X) :- sabroso(X) ; nutritivo(X).

5.

```
wizard(ron).
hasWand(harry).
quidditchPlayer(harry).
wizard(X):- hasBroom(X), hasWand(X).
hasBroom(X):- quidditchPlayer(X).
```

witch(ron).

1. Mago (Ron).
2. bruja (ron).
3. Mago (Hermione).
4. Bruja (Hermione).
5. Mago (Harry).
6. mago(Y).
7. bruja(Y).

Para empezar y realizar las consultas dadas, es necesario que todas estén en el mismo idioma que los hechos y reglas, porque si ponemos las reglas y hechos en inglés y queremos consultar en español, el editor dará error al no encontrar el predicado entre las cláusulas brindadas.

```
1 ?- Wizard (Ron).
ERROR: Syntax error: Operator expected
ERROR: Wizard
ERROR: ** here **
ERROR: (Ron) .
```

```
4 ?- witch (ron).
ERROR: Syntax error: Operator expected
ERROR: witch
ERROR: ** here **
ERROR: (ron) .
```

```
5 ?- Wizard (Hermione).
ERROR: Syntax error: Operator expected
ERROR: Wizard
ERROR: ** here **
ERROR: (Hermione) .
```

```
6 ?- Witch (Hermione).
ERROR: Syntax error: Operator expected
ERROR: Witch
ERROR: ** here **
ERROR: (Hermione) .
```

**Consulta 1.** Nos da error ya que tanto el predicado como su argumento están en mayúsculas, sin mencionar que hay un espacio entre ambos.

**Consulta 2.** En este caso, están bien escritos tanto el argumento como el funtor, pero también encontramos un espacio entre ambos.

**Consulta 3.** Pasa lo mismo que en el caso 1. Están separados por un espacio y ambos están en mayúsculas.

**Consulta 4.** Ídem consulta 1 y 3.

```
7 ?- Wizard (Harry).  
ERROR: Syntax error: Operator expected  
ERROR: Wizard  
ERROR: ** here **  
ERROR: (Harry) .
```

```
9 ?- wizard(Y).  
Y = ron .
```

```
10 ?- witch(Y).  
Y = ron.
```

**Consulta 5.** Ídem consultas 1, 3 y 4.  
**Consulta 6.** Aquí se usa Y como variable, y se busca qué valores hacen verdadera la consulta. Entre nuestros hechos, el único que cumple la condición es wizard(ron), por lo que tomará el valor del argumento y se lo asignará a la variable, dando como resultado Y = ron.

**Consulta 7.** Pasa lo mismo que en la consulta 6.

Para que las primeras consultas estén bien formuladas, deberíamos escribirlas de la siguiente forma:

```
3 ?- wizard(ron).  
true .
```

```
4 ?- witch(ron).  
true.
```

```
5 ?- wizard(hermione).  
false.
```

```
6 ?- witch(hermione).  
false.
```

```
7 ?- wizard(harry).  
true.
```