

ANIMALES

```
In [14]: !pip install clipspy
from clips import Environment, Symbol

environment = Environment()
```

Requirement already satisfied: clipspy in c:\users\narcisa\anaconda3\lib\site-packages (0.3.3)
Requirement already satisfied: cffi>=1.0.0 in c:\users\narcisa\anaconda3\lib\site-packages (from clipspy) (1.14.0)
Requirement already satisfied: pycparser in c:\users\narcisa\anaconda3\lib\site-packages (from cffi>=1.0.0->clipspy) (2.19)

```
In [15]: environment.load('animales.clp')
```

```
In [16]: environment.reset()
environment.run()
```

Out[16]: 10

```
In [17]: for rule in environment.rules():
          print(rule)
```

```
(defrule MAIN::mamifero-1
  (tiene-pelos)
=>
  (assert (es-mamifero)))

(defrule MAIN::mamifero-2
  (da-leche)
=>
  (assert (es-mamifero)))

(defrule MAIN::ungulado-1
  (es-mamifero)
  (tiene-pezuñas)
=>
  (assert (es-ungulado)))

(defrule MAIN::ungulado-2
  (es-mamifero)
  (rumia)
=>
  (assert (es-ungulado)))

(defrule MAIN::invertebrado-1
  (no-tiene-huesos)
=>
  (assert (es-invertebrado)))

(defrule MAIN::invertebrado-2
  (tiene-caparazon)
=>
  (assert (es-invertebrado)))

(defrule MAIN::artropodo-1
  (es-invertebrado)
  (tiene-patas)
=>
```

```
(assert (es-artropodo)))

(defrule MAIN::artropodo-2
  (es-invertebrado)
  (nacen-huevos)
  =>
  (assert (es-artropodo)))

(defrule MAIN::aracnidos-1
  (es-artropodo)
  (cuatro-pares-patas)
  =>
  (assert (es-aracnido)))

(defrule MAIN::aracnidos-2
  (es-artropodo)
  (no-tiene-antenas)
  =>
  (assert (es-aracnido)))

(defrule MAIN::insectos-1
  (es-artropodo)
  (tiene-tres-pares-patas)
  =>
  (assert (es-insecto)))

(defrule MAIN::insectos-2
  (es-artropodo)
  (tiene-antenas)
  =>
  (assert (es-insecto)))

(defrule MAIN::avez-1
  (tiene-plumas)
  =>
  (assert (es-ave)))

(defrule MAIN::avez-2
  (pone-huevos)
  =>
  (assert (es-ave)))

(defrule MAIN::gallinaceas-1
  (es-ave)
  (tiene-pico-corto)
  =>
  (assert (es-gallinacea)))

(defrule MAIN::gallinaceas-2
  (es-ave)
  (tiene-alas-cortas)
  =>
  (assert (es-gallinacea)))

(defrule MAIN::anceriformes-1
  (es-ave)
  (tiene-pico-aplanado)
  =>
  (assert (es-anceriforme)))

(defrule MAIN::anceriformes-2
  (es-ave)
  (tiene-patas-palmeadas)
  =>
  (assert (es-anceriforme)))
```

```
(defrule MAIN::corredoras-1
  (es-ave)
  (no-vuelan)
=>
  (assert (es-corredoras)))

(defrule MAIN::pato
  (es-anceriforme)
  (cuello-corto)
=>
  (assert (es-pato)))

(defrule MAIN::ganzo
  (es-anceriforme)
  (cuello-largo)
=>
  (assert (es-ganzo)))

(defrule MAIN::gallina
  (es-gallinacea)
  (es-domestica)
=>
  (assert (es-gallina)))

(defrule MAIN::codorniz
  (es-gallinacea)
  (tiene-alas-puntiagudas)
=>
  (assert (es-codorniz)))

(defrule MAIN::jirafa
  (es-ungulado)
  (tiene-cuello-largo)
=>
  (assert (es-jirafa)))

(defrule MAIN::cebra
  (es-ungulado)
  (tiene-rayas)
=>
  (assert (es-cebra)))

(defrule MAIN::abeja
  (es-insecto)
  (tiene-miel)
=>
  (assert (es-abeja)))

(defrule MAIN::mariposa
  (es-insecto)
  (no-pica)
=>
  (assert (es-mariposa)))

(defrule MAIN::arana
  (es-aracnido)
  (tiene-patas)
=>
  (assert (es-arana)))

(defrule MAIN::escorpion
  (es-aracnido)
  (tiene-pinzas)
=>
```

```
(assert (es-escorpion)))
```

```
In [18]: for fact in environment.facts():  
         print(fact)
```

```
(initial-fact)  
(tiene-pelos)  
(tiene-pezunias)  
(tiene-cuello-largo)  
(no-tiene-huesos)  
(nacen-huevos)  
(cuatro-pares-patas)  
(tiene-patas)  
(tiene-tres-pares-patas)  
(no-pica)  
f-10    (es-invertebrado)  
f-11    (es-artropodo)  
f-12    (es-aracnido)  
f-13    (es-arana)  
f-14    (es-insecto)  
f-15    (es-mariposa)  
f-16    (es-mamifero)  
f-17    (es-ungulado)  
f-18    (es-jirafa)
```