

In [1]:

```
1 #iNSTALAR CLIPSPY
2 pip install clipspy
```

```
Collecting clipspy
  Downloading clipspy-0.3.3-cp38-cp38-manylinux2010_x86_64.whl (68
6 kB)
    |████████████████████████████████████████| 686 kB 1.1 MB/s eta 0:00:0
1
Requirement already satisfied: cffi>=1.0.0 in /home/fernandosanche
z/anaconda3/lib/python3.8/site-packages (from clipspy) (1.14.0)
Requirement already satisfied: pycparser in /home/fernandosanchez/
anaconda3/lib/python3.8/site-packages (from cffi>=1.0.0->clipspy)
(2.20)
Installing collected packages: clipspy
Successfully installed clipspy-0.3.3
Note: you may need to restart the kernel to use updated packages.
```

In [1]:

```
1 from clips import Environment, Symbol
2 environment = Environment()
3 #Cargar El archivo dclp
4 environment.load('plantasReglas.clp')
5 #reiniciar Clips
6 environment.reset()
7 #ejecutar Clips
8 environment.run()
9 #imprimir las reglas
10 for rule in environment.rules():
11     print(rule)
12 #imprimit los fact
13 for fact in environment.facts():
14     print(fact)
```

```
(defrule MAIN::deficiencianitrogeno
  (La planta crece muy poco)
  =>
  (assert (La planta crece muy poco)))

(defrule MAIN::deficiencianitrogeno1
  (La planta tiene un color amarillo palido)
  =>
  (assert (La planta tiene un color amarillo palido)))

(defrule MAIN::deficiencianitrogeno2
  (Las hojas tienen un color pardo rojizo)
  =>
  (assert (Las hojas tienen un color pardo rojizo)))

(defrule MAIN::deficiencianitrogeno3
  (or (La planta crece muy poco)
      (La planta tiene un color amarillo palido)
      (Las hojas tienen un color pardo rojizo))
```

```
=>
(assert (nitrogeno))
(printout t "La planta tiene un deficiencia de nitrógeno." crlf
))

(defrule MAIN::deficienciafosforo
  (La raiz de la planta tiene poco crecimiento)
  =>
  (assert (La raiz de la planta tiene poco crecimiento)))

(defrule MAIN::deficienciafosforo1
  (La planta tiene un tallo fusiforme)
  =>
  (assert (La planta tiene un tallo fusiforme)))

(defrule MAIN::deficienciafosforo2
  (La planta tiene un color purpura)
  =>
  (assert (La planta tiene un color purpura)))

(defrule MAIN::deficienciafosforo3
  (La planta tiene retraso en la madurez)
  =>
  (assert (La planta tiene retraso en la madurez)))

(defrule MAIN::deficienciafosforo4
  (or (La raiz de la planta tiene poco crecimiento)
      (La planta tiene un tallo fusiforme)
      (La planta tiene un color purpura)
      (La planta tiene retraso en la madurez))
  =>
  (assert (fosforo))
  (printout t "La planta tiene un deficiencia de fósforo." crlf))

(defrule MAIN::deficienciapotasio
  (Los bordes de las hojas aparecen chamuscados)
  =>
  (assert (Los bordes de las hojas aparecen chamuscados)))

(defrule MAIN::deficienciapotasio1
  (Planta con los tallos debilitados)
  =>
  (assert (Planta con los tallos debilitados)))

(defrule MAIN::deficienciapotasio2
  (Planta con semillas o frutas marchitas)
  =>
  (assert (Planta con semillas o frutas marchitas)))

(defrule MAIN::deficienciapotasi4
  (or (Planta con los tallos debilitado)
      (Planta con los tallos debilitados)
      (Planta con semillas o frutas marchitas))
```

```

=>
(assert (potasio))
(printout t "La planta tiene un deficiencia de Potasio." crlf))

(initial-fact)

```

```

In [2]: 1 #Ingresar datos de la planta
        2 environment.reset()

```

```

In [3]: 1 desccripcion = input("Descripción de los síntomas de la planta
        2 problema = '(' + desccripcion + ')')
        3 #Afirma un hecho como una cadena.
        4 problema = environment.assert_string(problema)
        5 #imprimit los fact
        6 for fact in environment.facts():
        7     print(fact)

```

Descripción de los síntomas de la planta ? Planta con los tallos d  
ebilitado  
(initial-fact)  
(Planta con los tallos debilitado)

```

In [4]: 1 environment.run()
        2 #Imprimir resultado
        3 #print(environment.StdoutStream.Read())
        4

```

Out[4]: 1

```

In [5]: 1 for fact in environment.facts():
        2     print(fact)

```

(initial-fact)  
(Planta con los tallos debilitado)  
(potasio)

```

In [ ]: 1

```