

ExamenSE

December 23, 2020

1 UNIVERSIDAD POLITECNICA SALESIANA

```
[1]: from clips import Environment, Symbol
import tkinter

environment = Environment()
```

Procedemos a cargar nuestro sistema basado en reglas, para ello utilizamos el metodo load y le enviamos el archivo .clp

```
[2]: environment.load('examenSE.clp')
```

Procedemos a validar que se cargo el archivo de reglas con la siguiente instruccion

```
[3]: environment.reset()
environment.run()
```

```
[3]: 0
```

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

for fact in environment.facts():
    print(fact)
```

```
(defrule MAIN::nitrogeno1
  (La planta crece muy poco)
  =>
  (assert (La planta crece muy poco)))
```

```
(defrule MAIN::nitrogeno2
  (La planta tiene un color amarillo palido)
  =>
  (assert (La planta tiene un color amarillo palido)))
```

```
(defrule MAIN::nitrogeno3
  (La planta tiene hojas de color pardo rojizo)
  =>
```

```

(assert (La planta tiene hojas de color pardo rojizo)))

(defrule MAIN::fosforo1
  (La raiz de la planta crece poco)
=>
  (assert (La raiz de la planta crece poco)))

(defrule MAIN::fosforo2
  (La planta tiene tallo fusiforme)
=>
  (assert (La planta tiene tallo fusiforme)))

(defrule MAIN::fosforo3
  (La planta tiene color purpura)
=>
  (assert (La planta tiene color purpura)))

(defrule MAIN::fosforo4
  (La planta tiene retraso de madurez)
=>
  (assert (La planta tiene retraso de madurez)))

(defrule MAIN::potasio1
  (Tiene los bordes de las hojas chamuscados)
=>
  (assert (Tiene los bordes de las hojas chamuscados)))

(defrule MAIN::potasio2
  (La planta tiene tallos debilitados)
=>
  (assert (La planta tiene tallos debilitados)))

(defrule MAIN::potasio3
  (La planta tiene semillas o frutas marchitas)
=>
  (assert (La planta tiene semillas o frutas marchitas)))

(defrule MAIN::ausencia-nitrogeno
  (or (La planta crece muy poco)
       (La planta tiene un color amarillo palido)
       (La planta tiene hojas de color pardo rojizo))
=>
  (assert (no-nitrogeno))
  (printout t "La planta carece de Nitrógeno." crlf))

(defrule MAIN::ausencia-fosforo
  (or (La raiz de la planta crece poco)
       (La planta tiene tallo fusiforme)

```

```

        (La planta tiene color purpura)
        (La planta tiene retraso de madurez))
=>
(assert (no-fosforo))
(printout t "La planta carece de Fósforo." crlf))

(defrule MAIN::ausencia-potasio
  (or (Tiene los bordes de las hojas chamuscados)
       (La planta tiene tallos debilitados)
       (La planta tiene semillas o frutas marchitas))
=>
(assert (no-potasio))
(printout t "La planta carece de Potasio." crlf))

(initial-fact)

```

```

[5]: ventana = tkinter.Tk()
      ventana.geometry("500x300")

```

```

[5]: ''

```

```

[6]: environment.reset()
      sintomas = int(input("Numero de Sintomas:\n"))

      for sintoma in range(sintomas):
          deficiencia = input("Que problema tiene su planta ? ")
          fact_string_planta1 = '(' + deficiencia.capitalize() + ')'
          fact_planta1 = environment.assert_string(fact_string_planta1)

```

Numero de Sintomas:

2

Que problema tiene su planta ? la planta tiene color purpura

Que problema tiene su planta ? la planta tiene tallos debilitados

```

[ ]: environment.run()

      for fact in environment.facts():
          if fact.template.name == 'no-nitrogeno':
              etiqueta = tkinter.Label(ventana, text="La planta carece de Nitrógeno",
→fg="red").pack()
              print('La planta carece de Nitrógeno')
          elif fact.template.name == 'no-fosforo':
              etiqueta2 = tkinter.Label(ventana, text="La planta carece de Fósforo",
→fg="red").pack()
              print('La planta carece de Fósforo')
          elif fact.template.name == 'no-potasio':

```

```
etiqueta3 = tkinter.Label(ventana, text="La planta carece de Potasio",  
→fg="red").pack()  
print('La planta carece de Potasio')  
  
#etiqueta.pack()  
ventana.mainloop()
```

La planta carece de Potasio

La planta carece de Fósforo

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

```
[ ]:
```