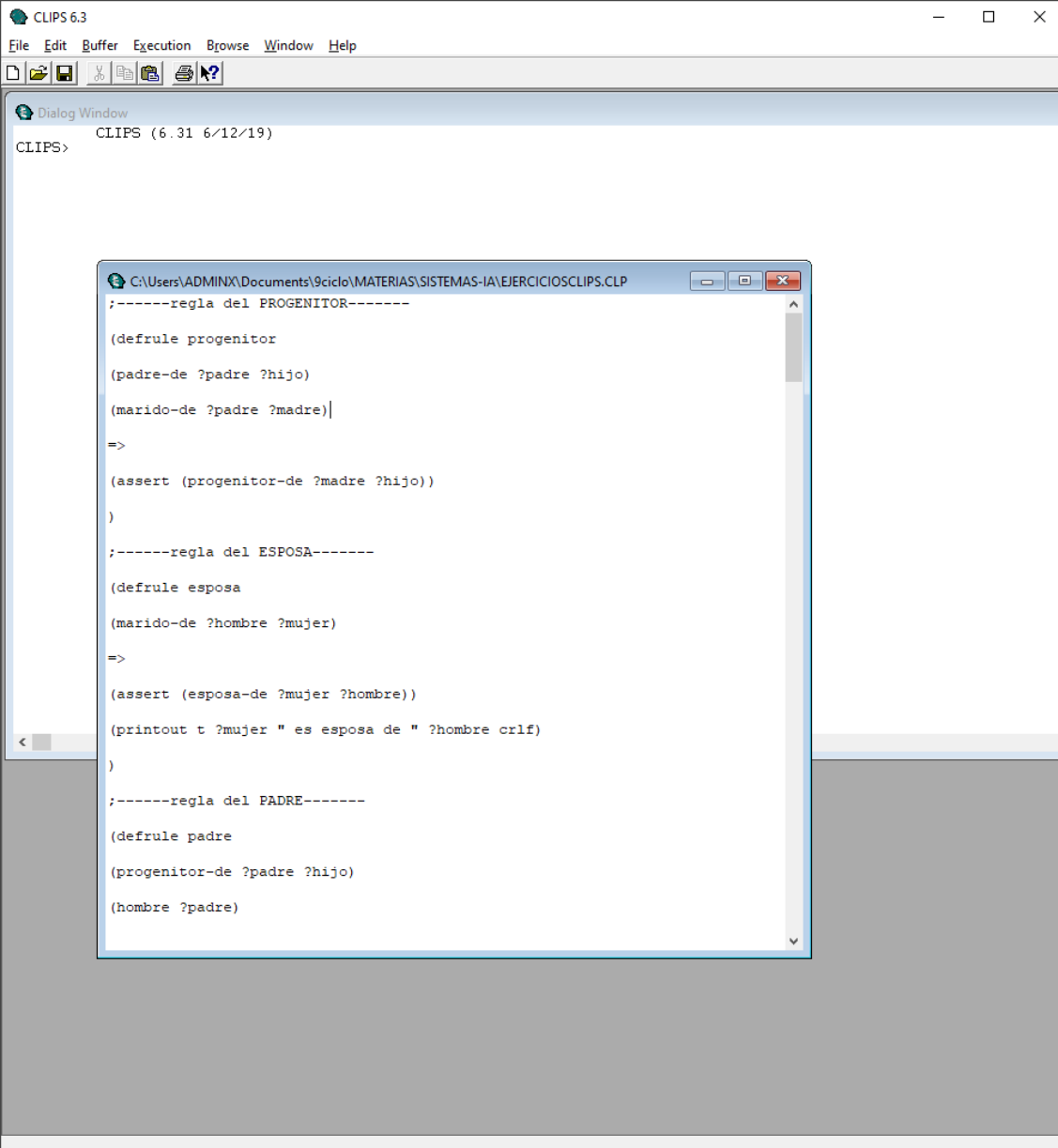


## EJERCICIOS DE CLIPS

Nombre: Vinicio Veletanga

### REGLAS DE SISTEMAS EXPERTOS



The screenshot shows the CLIPS 6.31 application window. The main window has a menu bar (File, Edit, Buffer, Execution, Browse, Window, Help) and a toolbar. A 'Dialog Window' is open, displaying the CLIPS prompt 'CLIPS>' and the file path 'CLIPS (6.31 6/12/19)'. Overlaid on this is a smaller text editor window titled 'C:\Users\ADMINX\Documents\9ciclo\MATERIAS\SISTEMAS-IA\EJERCICIOS\CLIPS.CLP'. This window contains three rule definitions in CLIPS syntax:

```
;-----regla del PROGENITOR-----  
  
(defrule progenitor  
  (padre-de ?padre ?hijo)  
  (marido-de ?padre ?madre)|  
=>  
  (assert (progenitor-de ?madre ?hijo))  
)  
  
;-----regla del ESPOSA-----  
  
(defrule esposa  
  (marido-de ?hombre ?mujer)  
=>  
  (assert (esposa-de ?mujer ?hombre))  
  (printout t ?mujer " es esposa de " ?hombre crlf)  
)  
  
;-----regla del PADRE-----  
  
(defrule padre  
  (progenitor-de ?padre ?hijo)  
  (hombre ?padre)
```

;-----regla del PROGENITOR-----

(defrule progenitor

(padre-de ?padre ?hijo)

(marido-de ?padre ?madre)

=>

(assert (progenitor-de ?madre ?hijo))

)

;-----regla del ESPOSA-----

(defrule esposa

(marido-de ?hombre ?mujer)

=>

(assert (esposa-de ?mujer ?hombre))

(printout t ?mujer " es esposa de " ?hombre crlf)

)

;-----regla del PADRE-----

(defrule padre

(progenitor-de ?padre ?hijo)

(hombre ?padre)

=>

(assert (padre-de ?padre ?hijo))

(printout t ?padre " es padre de " ?hijo crlf)

)

;-----regla del MADRE-----

(defrule madre

(progenitor-de ?madre ?hijo)

(mujer ?madre)

=>

```
(assert (madre-de ?madre ?hijo))
```

```
(printout t ?madre " es madre de " ?hijo crlf)
```

```
)
```

```
;-----regla del ABUELOS-----
```

```
(defrule abuelos
```

```
(progenitor-de ?padre ?hijo)
```

```
(progenitor-de ?hijo ?nieto)
```

```
=>
```

```
(assert (abuelos-de ?padre ?nieto))
```

```
)
```

```
;-----regla del ABUELO-----
```

```
(defrule abuelo
```

```
(abuelos-de ?padre ?nieto)
```

```
(hombre ?padre)
```

```
=>
```

```
(assert (abuelo-de ?padre ?nieto))
```

```
(printout t ?padre " es abuelo de " ?nieto crlf)
```

```
)
```

```
;-----regla del ABUELA-----
```

```
(defrule abuela
```

```
(abuelos-de ?madre ?nieto)
```

```
(mujer ?madre)
```

```
=>
```

```
(assert (abuela-de ?madre ?nieto))
```

```
(printout t ?madre " es abuela de " ?nieto crlf)
```

```
)
```

```
;-----regla del HERMANOS-PADRE-----
```

```
(defrule hermanos-padre
```

```
(padre-de ?padre ?hijo1)
```

```
(padre-de ?padre ?hijo2)
```

```
(test (neq ?hijo1 ?hijo2))
```

```
=>
```

```
(assert (hermanos ?hijo1 ?hijo2))
```

```
)
```

```
;-----regla del HERMANOS-MADRE-----
```

```
(defrule hermanos-madre
```

```
(madre-de ?madre ?hijo1)
```

```
(madre-de ?madre ?hijo2)
```

```
(test (neq ?hijo1 ?hijo2))
```

```
=>
```

```
(assert (hermanos ?hijo1 ?hijo2))
```

```
)
```

```
;-----regla del HERMANA-----
```

```
(defrule hermana
```

```
(hermanos ?hijo1 ?hijo2)
```

```
(mujer ?hijo1)
```

```
=>
```

```
(assert (hermana-de ?hijo1 ?hijo2))
```

```
(printout t ?hijo1 " es hermana de " ?hijo2 crlf)
```

```
)
```

```
;-----regla del HERMANO-----
```

```
(defrule hermano
```

```
(hermanos ?hijo1 ?hijo2)
```

```
(hombre ?hijo1)
```

```
=>
```

```
(assert (hermano-de ?hijo1 ?hijo2))
```

```
(printout t ?hijo1 " es hermano de " ?hijo2 crlf)
```

```
)
```

;-----completando-regla del TIO-----

(defrule tio

(hermanos ?hijo1 ?hijo2)

(hombre ?hijo2)

(progenitor-de ?hijo1 ?hijo)

=>

(assert (tio-de ?hijo2 ?hijo))

(printout t ?hijo2 " es tio de " ?hijo crlf)

)

;-----completando-regla del TIA-----

(defrule tia

(hermanos ?hijo1 ?hijo2)

(mujer ?hijo2)

(progenitor-de ?hijo1 ?hijo)

=>

(assert (tia-de ?hijo2 ?hijo))

(printout t ?hijo2 " es tia de " ?hijo crlf)

)

;-----completando-regla del SOBRINO-----

(defrule sobrino

(hermanos ?hijo1 ?hijo2)

(progenitor-de ?hijo1 ?hijo)

(hombre ?hijo)

=>

(assert (sobrino-de ?hijo ?hijo2))

(printout t ?hijo " es sobrino de " ?hijo2 crlf)

)

;-----completando-regla del SOBRINA-----

(defrule sobrina

(hermanos ?hijo1 ?hijo2)

(progenitor-de ?hijo1 ?hijo)

(mujer ?hijo)

=>

(assert (sobrina-de ?hijo ?hijo2))

(printout t ?hijo " es sobrina de " ?hijo2 crlf)

)

;-----completando-regla del PRIMO----- (defrule primo

(hermanos ?hijo1 ?hijo2)

(progenitor-de ?hijo1 ?hijo3)

(progenitor-de ?hijo2 ?hijo4)

(hombre ?hijo3)

=>

(assert (primo-de ?hijo3 ?hijo4))

(printout t ?hijo3 " es primo de " ?hijo4 crlf)

)

;-----completando-regla del PRIMA----- (defrule prima

(hermanos ?hijo1 ?hijo2)

(progenitor-de ?hijo1 ?hijo3)

(progenitor-de ?hijo2 ?hijo4)

(mujer ?hijo3)

=>

(assert (prima-de ?hijo3 ?hijo4))

(printout t ?hijo3 " es prima de " ?hijo4 crlf)

)

;-----FALTA REGLA SOBRE EL DIVORCIO-----

;-----Hechos-----

(defacts inicio

(hombre Diego)

(mujer Lili)

(hombre Juan)

(mujer Maria)

(hombre Luis)

(mujer Diana)

(hombre Mario)

(mujer Fanny)

(hombre Roberto)

(mujer Luisa)

(mujer Demi)

(progenitor-de Diego Demi)

(progenitor-de Luisa Diego)

(marido-de Diego Lili)

(progenitor-de Luisa Juan)

(progenitor-de Juan Diana)

(tio-de Juan Demi)

(sobrina-de Demi Juan)

(prima Demi Diana)

)