

# Clips: Variables

José A. Alonso y Francisco J. Martín

Ciencias de la Computación e Inteligencia Artificial

UNIVERSIDAD DE SEVILLA

# Variables en Clips (I)

- Sintaxis

```
variable ::= ?<nombre>
```

- Sesión

```
CLIPS> (clear)
CLIPS> (defrule regla-1
        (initial-fact)
        =>
        (printout t ?x crlf))
[PRCCODE3] Undefined variable x referenced
in RHS of defrule.
ERROR:
(defrule MAIN::regla-1
  (initial-fact)
  =>
  (printout t ?x crlf))
```

## Variables en Clips (II)

- Sesión

```
CLIPS> (assert (pelicula "Rojo"))
<Fact-0>
CLIPS> (defrule titulo
        (pelicula ?titulo)
        =>
        (assert (titulo ?titulo)))
CLIPS> (facts)
f-0      (pelicula "Rojo")
For a total of 1 fact.
CLIPS> (run)
CLIPS> (facts)
f-0      (pelicula "Rojo")
f-1      (titulo "Rojo")
For a total of 2 facts.
CLIPS>
```

# Regla con varias variables

- Sesión

```
CLIPS>(defrule quien-dirigio
      (director-de ?director ?pelicula)
      =>
      (printout t ?director " dirigio "
               ?pelicula crlf))

CLIPS> (reset)
CLIPS> (assert (director-de Kieslowski "Rojo"))
<Fact-1>
CLIPS> (agenda)
0      quien-dirigio: f-1
For a total of 1 activation.
CLIPS> (run)
Kieslowski dirigio Rojo
CLIPS> (assert (director-de Kieslowski "Azul"))
<Fact-2>
CLIPS> (run)
Kieslowski dirigio Azul
CLIPS> (assert (director-de Kieslowski))
<Fact-3>
CLIPS> (agenda)
CLIPS>
```

# Dirección de un hecho

- Sesión

```
CLIPS> (clear)
CLIPS> (assert (director-de Kieslowski "Rojo"))
<Fact-0>
CLIPS> (facts)
f-0      (director-de Kieslowski "Rojo")
For a total of 1 fact.
CLIPS> (defrule quien-dirigio
        ?hecho <- (director-de ?director ?pelicula)
        =>
        (printout t ?director " dirigio "
                 ?pelicula crlf)
        (retract ?hecho))
CLIPS> (run)
Kieslowski dirigio Rojo
CLIPS> (facts)
CLIPS>
```

- Comentario

- ?hecho <- (director-de ?director ?pelicula)

# Variables anónimas

- Sintaxis

variable-anonima ::= ?

- Sesión

```
CLIPS> (clear)
CLIPS> (defrule peliculas
  (director-de ? ?titulo)
  =>
  (printout t ?titulo crlf))
CLIPS> (assert (director-de Kieslowski "Rojo")
  (director-de Kieslowski "Blanco")
  (director-de Kieslowski "Octubre"
    por-confirmar))

<Fact-2>
CLIPS> (run)
Blanco
Rojo
CLIPS>
```

# Variables múltiples anónimas

- **Sintaxis**

```
variable-multiple ::= $?
```

- **Sesión**

```
CLIPS> (clear)
CLIPS> (defrule peliculas
  (director-de ? ?titulo $?)
  =>
  (printout t ?titulo crlf))
CLIPS> (assert (director-de Kieslowski "Rojo")
  (director-de Kieslowski "Blanco")
  (director-de Kieslowski "Octubre"
    por-confirmar))

<Fact-2>
CLIPS> (agenda)
0      peliculas: f-2
0      peliculas: f-1
0      peliculas: f-0
For a total of 3 activations.
CLIPS> (run)
Octubre
Blanco
Rojo
```

# Variables múltiples

- **Sintaxis**

```
variable-multiple ::= $? [<nombre>]
```

- **Sesión**

```
CLIPS>
CLIPS> (defrule peliculas
        (director-de ? ?titulo $?notas)
        =>
        (printout t ?titulo ?notas crlf))
CLIPS> (assert (director-de Kieslowski "Rojo")
          (director-de Kieslowski "Blanco")
          (director-de Kieslowski "Octubre"
            por-confirmar))

<Fact-2>
CLIPS> (agenda)
0      peliculas: f-2
0      peliculas: f-1
0      peliculas: f-0
For a total of 3 activations.
CLIPS> (run)
Octubre(por-confirmar)
Blanco()
Rojo()
CLIPS>
```



# Varias variables múltiples

- Sesión

```
CLIPS> (defacts cines
        (info sonido Alameda bueno)
        (info sonido Azul malo)
        (info sonido Arcos nuy-bueno)
        (info imagen Alameda buena)
        (info imagen Azul regular)
        (info imagen Arcos buena))
CLIPS> (reset)
CLIPS> (defrule infoalameda
        (info $?cabeza Alameda $?cola)
        =>
        (printout t "Cine Alameda: "
                  ?cabeza ?cola crlf))
CLIPS> (run)
Cine Alameda: (imagen)(buena)
Cine Alameda: (sonido)(bueno)
CLIPS>
```

# Manejo de listas

- Sesión

```
CLIPS> (assert (lista "Rojo" "Blanco" "Azul"))
<Fact-0>
CLIPS> (defrule lista
        ?hecho <- (lista $?cabeza "Blanco" $?resto)
        =>
        (retract ?hecho)
        (assert (imprime "Blanco" ?cabeza ?resto)))
CLIPS> (defrule impresion
        (imprime $?lista)
        =>
        (printout t ?lista crlf))
CLIPS> (run)
("Blanco" "Rojo" "Azul")
CLIPS>
```