

Tema 8: Clips: Plantillas

Ejemplo 1

- Sesión

```
CLIPS> (deftemplate datos
        "Datos de candidatos"
        (slot nombre
         (type STRING)
         (default ?DERIVE))
        (slot notas
         (type SYMBOL)
         (default rico))
        (slot edad
         (type NUMBER)
         (default 40)))
CLIPS> (assert (datos))
<Fact-0>
CLIPS> (assert (datos (edad 30) (nombre "Javier")))
<Fact-1>
CLIPS> (facts)
f-0      (datos (nombre "") (notas rico) (edad 40))
f-1      (datos (nombre "Javier") (notas rico)
              (edad 30))
For a total of 2 facts.
CLIPS>
```

Ejemplo con restricciones

- Sesión

```
CLIPS> (deftemplate grados
        (slot grado
          (allowed-symbols a b c d f))
        (slot escala
          (range 1 10)))
CLIPS> (assert (grados))
<Fact-1>
CLIPS> (facts)
f-0      (initial-fact)
f-1      (grados (grado a) (escala nil))
For a total of 2 facts.
CLIPS> (assert (grados (grado a) (escala 5)))
<Fact-2>
CLIPS> (facts)
f-0      (initial-fact)
f-1      (grados (grado a) (escala nil))
f-2      (grados (grado a) (escala 5))
For a total of 3 facts.
CLIPS> (assert (grados (escala 11)))
[CSTRNCHK1] A literal slot value found in the
assert command does not fall in the allowed
range 1 to 10 for slot escala.
CLIPS> (assert (grados (grado e)))
[CSTRNCHK1] A literal slot value found in the
assert command does not match the allowed
values for slot grado.
CLIPS>
```

Acceso y uso de plantillas

- Sesión

```
CLIPS> (deftemplate datos
  "Datos de candidatos"
  (multislot nombre
    (type STRING)
    (default ?DERIVE))
  (slot notas
    (type SYMBOL)
    (default rico))
  (slot edad
    (type NUMBER)
    (default 40)))
CLIPS> (defrule candidato-matrimonial
  (datos (nombre $?nombre)
    (notas ?notas)
    (edad ?edad))
  =>
  (printout t "Nombre: " ?nombre crlf
    "Notas: " ?notas crlf
    "Edad: " ?edad crlf))
CLIPS> (assert (datos (edad 30) (nombre "Javier")))
<Fact-0>
CLIPS> (run)
Nombre: ("Javier")
Notas: rico
Edad: 30
CLIPS>
```

Manipulación de plantillas

● Sesión

```
CLIPS> (defrule elimina-candidatos
        ?candidato <- (datos (notas pobre)
                             (nombre $?nombre))
        =>
        (retract ?candidato)
        (printout t "Candidato eliminado: "
                  ?nombre crlf))
CLIPS> (assert (datos (nombre "Fernando" "Martinez")
                     (notas pobre)))
<Fact-1>
CLIPS> (assert (datos (nombre "Martin" "Fernandez")
                     (notas rico)))
<Fact-2>
CLIPS> (facts)
f-0      (initial-fact)
f-1      (datos (nombre "Fernando" "Martinez")
             (notas pobre) (edad 40))
f-2      (datos (nombre "Martin" "Fernandez")
             (notas rico) (edad 40))
For a total of 3 facts.
CLIPS> (run)
Candidato eliminado: ("Fernando" "Martinez")
CLIPS> (facts)
f-1      (datos (nombre "Martin" "Fernandez")
             (notas rico) (edad 40))
For a total of 1 fact.
CLIPS>
```

Un ejemplo más complejo

- Sesión

```
CLIPS> (deftemplate datos
  "Datos de candidatos"
  (multislot nombre
    (type STRING)
    (default ?DERIVE))
  (slot notas
    (type SYMBOL)
    (allowed-symbols pobre rico medio)
    (default rico))
  (slot edad
    (type NUMBER)
    (range 28 ?VARIABLE)
    (default 40)))
CLIPS> (assert (datos (nombre "Fernando" "Martinez")
  (notas pobre)))
<Fact-1>
CLIPS> (defrule elimina-candidatos
  ?candidato <- (datos (notas pobre)
    (nombre $?nombre))
  =>
  (retract ?candidato)
  (printout t "Candidato eliminado: "
    ?nombre crlf))
CLIPS> (run)
Candidato eliminado: ("Fernando" "Martinez")
CLIPS>
```

Modificación de plantillas

- Sesión

```
CLIPS> (undefrule *)
CLIPS> (defrule empobrecer
  ?datos <- (datos (nombre $?nombre)
              (notas rico)
              (edad ?edad))
  =>
  (printout t "Nombre: " ?nombre crlf
             "Notas: " notas crlf
             "Edad: " ?edad crlf)
  (modify ?datos (notas pobre)))
CLIPS> (assert (datos (edad 30) (nombre "Javier")))
<Fact-1>
CLIPS> (facts)
f-1      (datos (nombre "Javier")
           (notas rico) (edad 30))
For a total of 1 fact.
CLIPS> (run)
Nombre: ("Javier")
Notas:  notas
Edad:   30
CLIPS> (facts)
f-2      (datos (nombre "Javier")
           (notas pobre) (edad 30))
For a total of 1 fact.
CLIPS>
```

Sintaxis de plantillas (I)

```
(deftemplate <nombre> [<comentario>]
  <definicion-de-ranura>*)
```

```
<definicion-de-ranura>
  ::= <definicion-de-ranura-simple> |
     <definicion-de-ranura-multiple>
```

```
<definicion-de-ranura-simple>
  ::= (slot <nombre-de-ranura>
       <atributo-de-ranura>*)
```

```
<definicion-de-ranura-multiple>
  ::= (multislot <nombre-de-ranura>
       <atributo-de-ranura>*)
```

```
<atributo-de-ranura>
  ::= <atributo-defecto> |
     <restriccion-atributo>
```

```
<atributo-defecto>
  ::= (default ?DERIVE | ?NONE)
```


Sintaxis de plantillas (II)

```
<restriccion-atributo>
 ::= (type <tipo>) |
      <atributos-permitidos> |
      <rango-de-atributo>

<tipo>
 ::= SYMBOL | STRING | NUMBER |
      INTEGER | FLOAT

<atributos-permitidos>
 ::= (allowed-symbols <simbolo>+) |
      (allowed-strings <cadena>+) |
      (allowed-numbers <numero>+) |
      (allowed-integers <entero>+) |
      (allowed-floats <flotante>+) |
      (allowed-values <constante>+)

<rango-de-atributo>
 ::= (range <inicial> <final>)

<inicial> ::= <numero>
<final>   ::= <numero>
```