Módulos

El constructor *defmodule*

Introducción

- CLIPS permite estructurar las base de conocimientos y al mismo tiempo facilitar el control del razonamiento de forma modular.
- **defmodule** agrupa constructores (hechos, reglas, etc.) en módulos.

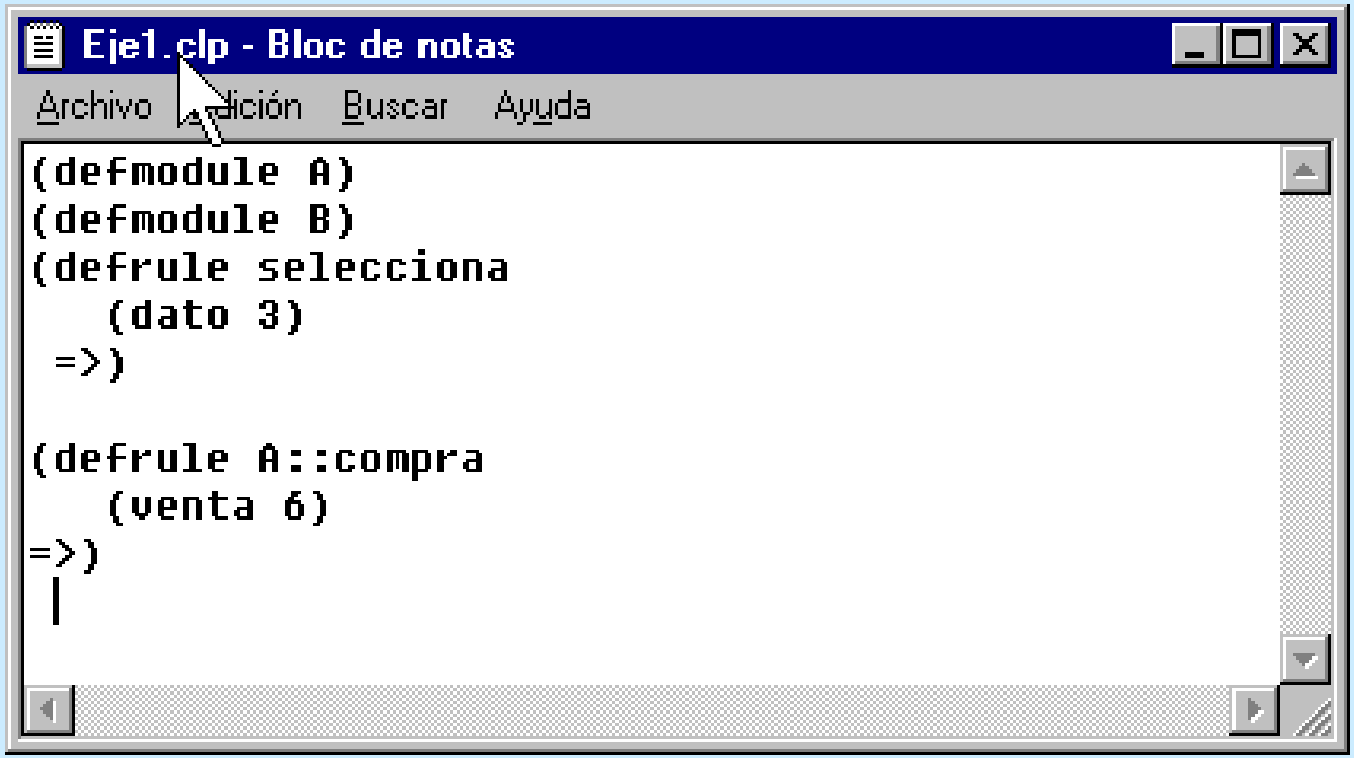
Sintaxis

```
(defmodule <nombre-del-módulo> [<commentario>]
   <especificación-portabilidad>*)

<especificación-portabilidad> ::=
                    (export <port-item>) |
                    (import <module-name> <port-item>)

<port-item>  ::= ?ALL |
                 ?NONE |
        <constructor-portable> ?ALL |
        <constructor-portable> ?NONE |
        <constructor-portable> <nombre-constructor>+

<constructor-protable>    ::= deftemplate | defclass
  |defglobal | deffunction | defgeneric
```

Definición de módulos

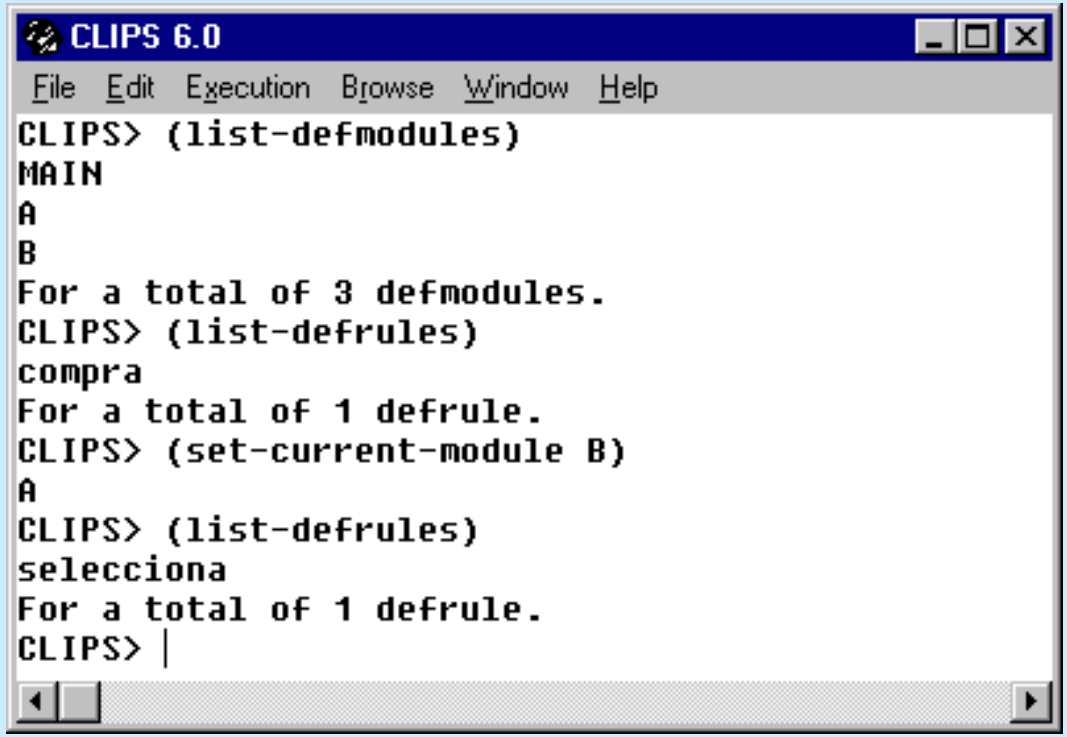
- La única forma de borrar un módulo es con la orden ***clear***.
- Siempre existe un módulo MAIN.
- Los módulos asociados con un nombre pueden especificarse explícitamente (modulo::elemento) o implícitamente (módulo actual o módulo activo).

Definición de módulos



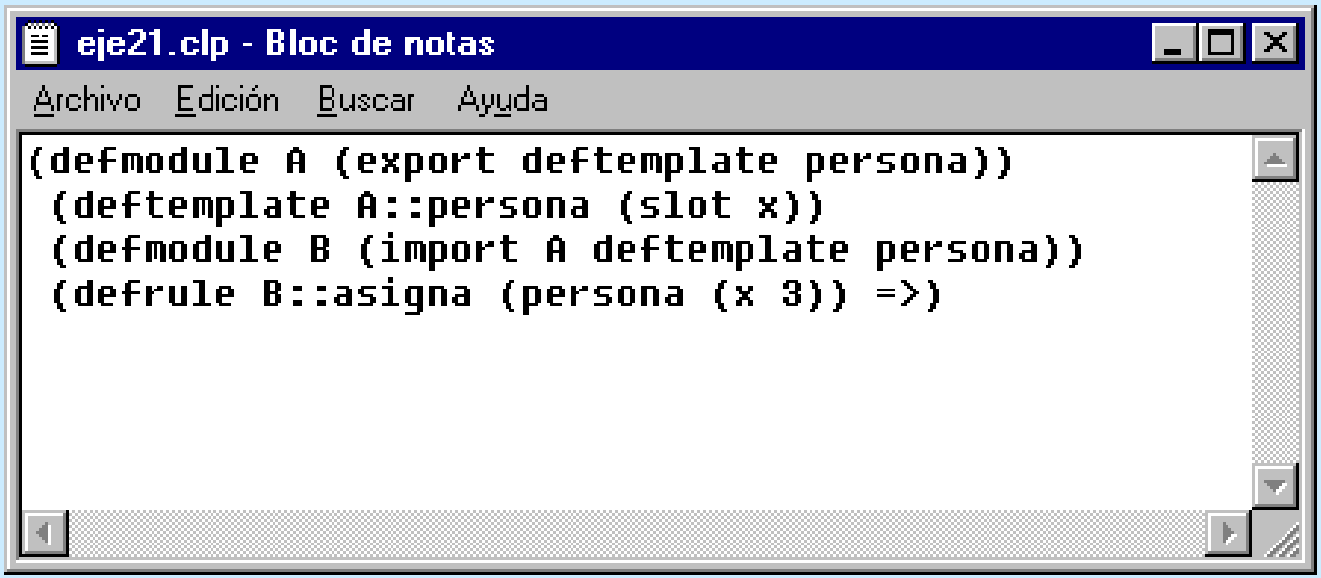
Definición de Módulos

- La orden **set-current-module** activa un módulo diferente.



Importando y exportando elementos

- Los elementos deben ser exportado o importado para poder ser usados por otros módulos.



Ejecución de Reglas

- Cada módulo tiene su agenda.
- ***(run)*** ejecuta la agenda del módulo activo .
- ***(reset)*** y ***(clear)*** hacen que el módulo MAIN tenga el foco actual.
- Existe una pila de módulos activos y se van ejecutando secuencialmente.
- Cada módulo ejecuta sus propias reglas de su agenda.
- Un programa termina su ejecución si no quedan módulos activos.

Ejecución de Reglas

- Formas de alterar la ejecución normal:
  - Orden **focus** en el consecuente de una regla. Incluye un nuevo módulo en la pila de módulos activos y pasa el control a la agenda de este módulo.
  - Orden ***return*** en el consecuente de una regla. Esto provocaría que se pasaría como módulo activo el siguiente módulo de la pila de módulos activos y pasa el control a su agenda.

Ejecución de Reglas

Eje3.clp - Bloc de notas

Archivo Edición Buscar Ayuda

```
(defmodule MAIN (export ?ALL))
(deftemplate MAIN::objetivo (slot nombre) (slot estado))
(defrule MAIN::ejemplo-foco
  =>
   (printout t "Disparo en el modulo MAIN." crlf)
   (assert (objetivo (nombre principal) (estado resuelto)))
   (focus A B))
(defmodule A
        (export  ?ALL)
        (import MAIN deftemplate initial-fact objetivo))
(deftemplate A::subobjetivo-A (slot nombre) (slot estado))
(defrule A::regla-ejemplo
(objetivo (nombre principal) (estado resuelto))
=>
   (assert (subobjetivo-A (estado resuelto)))
   (printout t "Disparando la regla en el modulo A." crlf))
(defmodule B (import A    deftemplate subobjetivo-A)
             (import MAIN deftemplate initial-fact) )
(defrule B::regla-ejemplo
  (subobjetivo-A (estado resuelto))
  =>
  (printout t "Disparando la regla en el modulo B." crlf))
```