

# Instalar CLIPS

<https://sourceforge.net/projects/clipsrules/files/CLIPS/6.40/>

## En windows

- clips\_windows\_[64/32]\_bit\_installer\_[version].msi
- Ejecutar el instalador normalmente
- Abrir ClipsIDE

## En linux

- clips\_core\_source\_[version].zip
- Extraer la carpeta del comprimido (unzip clips\_core\_source\_631.zip)
- cd clips\_core\_source\_631/core/
- sudo make
- ./clips

# Programación en CLIPS

Es un lenguaje que corre en shell, es decir, que posee una terminal en la que escribir comandos sencillos:

**(+ 3 4)**

**(persona**

**(color-pelo castaño)**

**(nombre Francisco)**

**(edad 23)**

**)**

**(retract)**

**(facts)**

**(assert)**

**(clear)**

**(exit)**

# Programación en CLIPS

## Archivo .clp

- Se puede codificar en el IDE o en cualquier procesador de texto que permita exportar la extensión .clp (bloc de notas por ejemplo)
- Se compone de Hechos (Facts) y Reglas (Rules): La programación consiste en definir el conjunto de reglas para que cuando se introduzcan los hechos el sistema navegue por el árbol de decisión.

# Programación en CLIPS

Defrule:

- Permite definir reglas como secuencias IF-THEN:

```
(defrule Regla1
  =>
  (assert (hecho1 ok))
)
```

```
(defrule Regla2
  (hecho1 ?)
  =>
  (assert (hecho2 ok))
)
```

```
(defrule Regla3
  (hecho2 ?h2)
  =>
  (if (eq h2 ok)
    then
      (assert (hecho2 ok))
  )
)
```

# Programación en CLIPS

```
(defrule Regla4
```

```
  =>
```

```
  (printout t "Pido algo por consola: ")
```

```
  (assert (hecho4 (read))))
```

```
)
```

```
(defrule Regla5
```

```
  (hecho4 ? h4)
```

```
  =>
```

```
  (if (eq h4 No-me-importa)
```

```
  then
```

```
    (printout t " :'( ")
```

```
  )
```

```
)
```

# Ejecución en CLIPS

**(load nombre-del-archivo.clp)**

**(reset)**: limpia los hechos asumidos por el programa, pero no las reglas

**(clear)**: limpia hechos y reglas

**(run)**: ejecuta el programa