

FUNCIONES DE BIFURCACIÓN	OBSERVACIONES
<p><b>(cond (prueba1 resultado1 ...) ...)</b></p> <p><i>Es la función condicional multirrama de LISP, equivalente al CASE de otros lenguajes</i></p>	<ul style="list-style-type: none"> <li>• La función cond acepta cualquier número de listas como argumentos. Evalúa el primer elemento de cada lista (en el orden indicado) hasta que uno de ellos devuelva un valor distinto de nil. A continuación, evalúa las expresiones que siguen a este elemento y devuelve el valor de la última expresión de la sublista. Si esta sublista sólo contiene un valor (es decir, si falta resultado), se obtiene el valor de la expresión prueba.</li> <li>• Cond se puede utilizar como una función de tipo case. Es habitual utilizar T como última (por defecto) expresión de prueba.</li> </ul>
<p><b>(if expr_prueba expr_entonces [expr_sino])</b></p> <p><i>Es la bifurcación clásica de cualquier lenguaje de programación</i></p>	<ul style="list-style-type: none"> <li>• Si expr_prueba no es nil, evalúa expr_entonces; en caso contrario evalúa expr_sino. La función if devuelve el valor de la expresión seleccionada. Si expr_luego no existe y expr_prueba es nil, entonces la función if devuelve nil.</li> <li>• Tanto la salida entonces como la salida sino contienen exclusivamente una lista. En caso de existir varias declaraciones en una u otra salida, hay que usar la función progn.</li> </ul>
<p><b>(progn [expr] ...)</b></p> <p><i>Calcula las expresiones secuencialmente y devuelve el valor de la última expresión</i></p>	

## EJEMPLOS DE FUNCIONES DE BIFURCACIÓN

```
(setq a 25 b 22)
22
(if (> a b)
    (setq a (+ a 10)); Then
    (setq b (+ b 10)); Else
```

```
)
35
(setq a 25 b 22)
22
(if (> a b)
    (progn
      (setq a (+ a 10))
      (print b)
    ); Then
    (setq b (+ b 10)); Else
```

```
)
; El fragmento valora a = 35 y
; presenta el valor 22 por pantalla
```

```
(setq a 2 b 4 c 6 m 4)
4
(cond
  ((= m a) (setq m (+ m 1)))
  ((= m b) (setq m (+ m 3)))
  ((= m c) (setq m (+ m 5)))
  (T (setq m 0))
```

```
)
7
```

;La función tiene cuatro ramas y al ejecutarse, sale por la segunda haciendo que la variable m valga 4 + 3. Si las tres ramas iniciales dan suceso falso, se llega a la cuarta que, al tener explícitamente la premisa T, siempre se ejecuta. Esta es la forma de proponer la condición ***“en caso de que las anteriores condiciones no se cumplan”***.