



1. (CAR <lista>) o bien **FIRST** Ej.: (CAR '(1 2 3)) → 1
2. (CDR <lista>) o bien **REST** Ej.: (CDR '(1 2 3)) → (2 3)
3. (NTH <posición> <lista>) Ej.: (NTH 2 '(a b c d)) → c
4. (CONS <obj> <lista>) Ej.: (CONS 'a '(1 2 3)) → (a 1 2 3)
5. (APPEND <lista> <lista>) Ej.: (APPEND '(a b) '(1 2 3)) → (a b 1 2 3)
6. (LIST <obj₁> <obj₂> ... <obj_n>) Ej.: (LIST 'a 'b '(1 2 3)) → (a b (1 2 3))
7. (ATOM <obj>) Ej.: (ATOM 'a) → T / (ATOM '(a b)) → NIL / (ATOM '(a)) → NIL
8. (SYMBOLP <obj>) Ej.: (SYMBOLP 'a) → T / (SYMBOLP '(a)) → NIL / (SYMBOLP '5) → NIL
9. (NUMBERP <obj>) Ej.: (NUMBERP '5) → T / (NUMBERP '(5)) → NIL / (NUMBERP '(a)) → NIL
10. (LISTP <obj>) Ej.: (LISTP '(a b)) → T / (LISTP 'a) → NIL / (LISTP '()) → T
11. (NULL <obj>) Ej.: (NULL '(a b)) → NIL / (NULL 'a) → NIL / (NULL '()) → T
12. (LENGTH <lista>) Ej.: (LENGTH '(a b c)) → 3 / (LENGTH '(a (b c) d)) → 3 / (LENGTH '()) → 0
13. (IF <expr₁> <expr₂> <expr₃>) El valor del condicional IF es el valor de la <expr₂> si el valor de <expr₁> es T o bien el valor de la <expr₃> si el valor de la <expr₁> es NIL.
14. (COND (<expr₁₁> <expr₁₂>)
(<expr₂₁> <expr₂₂>)
...
(<expr_{n1}> <expr_{n2}>)
)
El valor del condicional COND será:
el valor de la <expr₁₂> si el valor de la <expr₁₁> es T
el valor de la <expr₂₂> si el valor de la <expr₂₁> es T y <expr₁₁> es NIL
...
el valor de la <expr_{n2}> si el valor de la <expr_{n1}> es T y todas las <expr₁₁> anteriores son NIL
15. (DEFUN <nombre> (<parámetros> &optional (<parám op> <valor>) ... (<parám op> <valor>))
<cuerpo>) parámetro opcionales con valores por defecto.
16. Funciones anónimas (LAMBDA <lista parámetros> <cuerpo>) Ej.: ((LAMBDA (X) (+ X 1))
5) → 6
17. (MAPCAR <función> <lista>) Consiste en aplicarle la función a todos los elementos de la lista
obteniendo una nueva lista. EJ: (MAPCAR atom '(a nil (1 2))) → (t t nil)
18. (APPLY <función> <lista>)
Si la función que se especifica es monádica, la lista debe contener un solo elemento, y el resultado es el valor
que devuelve la función aplicada a ese elemento.
Si la función que se especifica es binaria, la lista debe contener solo dos elementos, y el resultado es el valor
que devuelve la función aplicada a esos dos elementos.
Y así para funciones de más argumentos.
Ej.: (APPLY car '(a b)) → a Ej.: (APPLY cons '(a (1 2))) → (a 1 2)