FUNICIONES CONSTRUCTORAS DE LISTAS

CONS: Tiene sólo 2 argumentos. El resultado final es una lista que tiene como 1er. elemento (CAR) toda la estructura del 1er argumento (átomo o lista) y como resto (CDR) todos los elementos del 2do. argumento detectados en el primer nivel. En otras palabras: El 1er argumento pasa a ser el 1er. elemento con toda su estructura (átomo o lista) del 2do argumento . Cuando el 2do. argumento es un átomo genera una lista punteada.

[un átomo o una lista] Si lo que se desea es generar una lista con solamente un elemento (átomo o lista), por ejemplo: A; (A); ((A B C)), se agrega como último argumento NIL (que es átomo y lista a la vez).

> (cons 'A nil) **(A)** > (cons '(A) nil) **((A))**

> (cons '((A B C)) ()) **(((A B C)))**

[átomo átomo] > (cons 'A 'B) **(A . B)** > (cons '1 2) **(1 . 2)** > (cons 'A 2) **(A . 2)**

[lista átomo] > (cons '(A B) 'X) **((A B) . X)**

> (cons nil 2) **(NIL . 2)** > (cons 'nil 'A) **(NIL . A)**

[átomo lista] > (cons 'X '(F G)) **(X F G)** > (cons 8 '((F G))) (**8 (F G))**

[lista lista] > (cons '(A B) '(1 2)) **((A B) 1 2)**

LIST : Tiene infinitos argumentos (átomos o listas). Genera una **lista propia** donde cada elemento es cada uno de los argumentos con su estructura original (átomo o lista).

[un átomo o una lista] > (list 'A) **(A)** > (list '(A B)) **((A B))**

[todos átomos] > (list 'A 'B) **(A B)** > (list 'A 2) **(A 2)**

[todos listas] > (list '(1 2) '((A B)) '(9)) **((1 2) ((A B)) (9))**

[combinación de átomos y listas] > (list 'A '(Z Z)) **(A (Z Z))**

> (list 'A '((Z Z)) 45 '(A ((B B)))) **(A ((Z Z)) 45 (A ((B B))))**

APPEND : Tiene infinitos argumentos. Todos los argumentos deben ser listas excepto el último que puede ser átomo. Genera una lista que tiene como elementos, los elementos detectados en el 1er. nivel de cada uno de los argumentos (excepto que su único argumento sea un átomo, caso éste en la que NO genera lista). Si el último argumento es un átomo genera una lista punteada.

[un átomo o una lista] > (append 'A) **A** > (append '(A B)) **(A B)**

[todos listas] > (append '(AB) '((X Y))) **(AB (X Y))**

> (append '((1 2 3)) '(Z) ' (A B)) **((1 2 3) Z A B)**

[lista lista .. átomo] > (append '(A B) '((C D)) 'E) **(A B (C D) . E)**

> (append '(A B) '((C D)) '9) **(A B (C D) . 9)**

> (append '(A B) '(C D) 'E) **(A B C D . E)**