

Ejercicio 1:

Da otro Calculo posible para el resultado de double (double 2)

double (double 2)

= {Aplicando el double externo}

double 2 + double 2

= {Aplicando ambos double }

(2 + 2) + (2 + 2)

= { aplicando los + interiores}

4 + 4

= {Aplicando el + }

8.

Ejercicio 2:

Muestra que  $\text{sum } [x] = x$  para cualquier número x.

$\text{Sum}[x]$

= {aplicando sum}

$X + \text{sum}[]$

= {aplicando sum}

$X + 0$

= {aplicando +}

X

Ejercicio 3:

Define una función product que produzca el producto de una lista de números y muestra usando tu definición product  $[2,3,4] = 24$ .

Definición:

$\text{product } [] = 1$

$\text{product } [n:ns] = n * \text{product } ns$

entonces:

$\text{product } [2,3,4]$

= {aplicando product}

$2 * \text{product } [3,4]$

= {aplicando product}

2 \* ( 3 \* product [4])

= {aplicando product}

2 \* ( 3 \* ( 4 \* product [] ))

= {aplicando product}

2 \* (3\* (4 \*1))

= {aplicando \*}

24.

#### Ejercicio 4:

¿Cómo debería modificarse la definición de la función qsort para que produzca una versión ordenada inversamente de una lista?

qsort [] = []

qsort (x:xs) = qsort larger ++ [x] ++ qsort smaller

donde:

smaller = [a | a <- xs, a <= x]

larger = [b | b <- xs, b > x]

¿Cuál sería el efecto de reemplazar <= por < en la definición original de qsort? Sugerencia: considera el ejemplo qsort[2,2,3,1,1].

En este caso se perdería un elemento 2 y un elemento 1:

qsort[2,2,3,1,1]

=

qsort[1,1] ++ [2] ++ qsort[3]

=

qsort[] ++ [1] ++ qsort[] ++ [2] ++ qsort[] ++ [3] ++ qsort[]

= [1,2,3].