

**Programación Declarativa (Prog. Funcional)**

APELLIDOS: \_\_\_\_\_

NOMBRE: \_\_\_\_\_

	1	2	3	4	5	6	total
PUNTOS:	1.0	1.0	1.0	1.0	3.0	3.0	10.0

☐ si } deseo que se publique mi calificación si fuera negativa  
☐ no }

Días de asistencia a clase en este parcial:  de 8

Consideremos las funciones estándares de Prelude:

$dropWhile\ p\ [] = []$	$[] ++\ ys = ys$
$dropWhile\ p\ (x : xs)$	$(x : xs) ++\ ys = x : (xs ++\ ys)$
$ p\ x = dropWhile\ p\ xs$	$length\ [] = 0$
$ otherwise = x : xs$	$length\ (x : xs) = 1 + length\ xs$

**1** Escribe los tipos de las funciones anteriores: $dropWhile ::$  $length ::$  $(++) ::$ **2** Evalúa la FN (forma normal) de las expresiones $dropWhile\ (\leq\ 2)\ [1, 2, 3, 4, 5, 6, 7] \rightsquigarrow$  $head\ (dropWhile\ (\leq\ n)\ [0..]) \rightsquigarrow$ **3** ¿Qué valor computa la expresión  $head\ (dropWhile\ p\ [0..])$ ?**4** Definimos la raíz cúbica de un número natural  $n$  como el menor natural  $x$  que satisface  $n < (x+1)^3$ . Utilizando  $dropWhile$  escribe una función para calcular la raíz cúbica de un natural: $raízCúbica :: Integer \rightarrow Integer$  $raízCúbica\ n =$ **5** Sean  $xs, ys, zs$  tres listas en FN (forma normal) con longitudes  $lx, ly$  y  $lz$  respectivamente. Prueba (por inducción) que para reducir lista la  $xs ++ ys$  a FN son necesarias  $lx + 1$  reducciones.**6** ¿Cuántas reducciones son necesarias para calcular la FN de  $(xs ++ ys) ++ zs$ ?¿Y para calcular la FN de  $xs ++ (ys ++ zs)$ ?¿Es mejor la declaración (A)  $infixr\ 5\ ++$  que la declaración (B)  $infixl\ 5\ ++$ ?