

Ejercicios Prácticos de Programación Declarativa

Sesión de laboratorio 2

Curso 2021/22

- Realiza los siguientes ejercicios individualmente en un mismo fichero .hs.
 - Escribe tu nombre al comienzo del fichero como líneas comentadas. Incluye comentarios significativos y no olvides declarar los tipos de las expresiones que defines.
1. Programa en Haskell las siguientes funciones cuyo argumento es un entero no negativo n , sin usar recursión explícita sino **funciones de orden superior** predefinidas en Haskell.
 - a) La lista de los cuadrados de los números naturales entre 0 y n (o sea, $[0, 1, 4, 9, \dots, n^2]$).
 - b) La lista anterior, pero con cada número emparejado con su cuadrado y en orden inverso $([(n, n^2), \dots, (2, 4), (1, 1), (0, 0)])$.
 - c) La suma $\sum_{i=1}^{i=n} i \cdot | \cos(i) |$.
 - d) La suma de los números menores que n que sean múltiplos de 3 o 5.
 - e) El primer número primo mayor que n .
 2. Programa las siguientes funciones de orden superior, utilizando funciones de orden superior predefinidas en Haskell. Los tipos de las variables n y m usadas en estas funciones tienen que estar en la clase **Enum**.
 - a) **iguales** $f\ g\ n\ m \Leftrightarrow f\ x = g\ x$, para todo $n \leq x \leq m$.
 - b) **menor** $n\ p =$ menor $x \geq n$ que verifica p
 - c) **mayorA** $n\ m\ p =$ mayor x con $n \leq x \leq m$ que verifica p
 - d) **ex** $n\ m\ p \Leftrightarrow$ existe x con $n \leq x \leq m$ que verifica p .
 3. Programa las siguientes funciones de orden superior, utilizando funciones de orden superior predefinidas en Haskell:
 - a) **filter2** $xs\ p\ q = (us, vs)$ donde us son los elementos de xs que cumplen p y vs los que cumplen q .
 - b) **filters** $xs\ ps = [xs_1, \dots, xs_n]$, donde xs_i son los elementos de xs que cumplen p_i , supuesto que ps es $[p_1, \dots, p_n]$.
 - c) **mapx** $x\ [f_0, f_1, \dots, f_n] = [f_0\ x, f_1\ x, \dots, f_n\ x]$.
 4. Programa, indicando los tipos, las siguientes variantes de **foldl** y **foldr**, que operan con listas no vacías y no usan valor acumulado inicial:
 - **foldr1** $\oplus\ [x_1, \dots, x_n] = x_1 \oplus x_2 \oplus \dots \oplus x_n$ (con \oplus asociando por la derecha)
 - **foldl1** $\oplus\ [x_1, \dots, x_n] = x_1 \oplus x_2 \oplus \dots \oplus x_n$ (con \oplus asociando por la izquierda)