

# Ejercicios Prácticos de Programación Declarativa

## Sesión de laboratorio 1

Curso 2020/21

- Realiza los siguientes ejercicios en un mismo fichero `.hs`.
  - **Escribe tu nombre al comienzo del fichero como líneas comentadas.**
  - Incluye comentarios explicativos.
  - Sube el fichero con tus soluciones al Campus Virtual.
1. Define expresiones Haskell para los siguientes apartados:
    - a) Calcular cuántos años hay en  $10^6$  segundos (supón que todos los años tienen 365 días; en otro momento puedes hacerlo teniendo en cuenta bisiestos).
    - b) Calcular cuántos años enteros, días restantes enteros, horas restantes enteras, minutos restantes enteros y segundos restantes hay en  $10^6$  segundos.
    - c) Generaliza los dos apartados anteriores, convirtiendo el número de segundos en argumento de una función. Declara los tipos de las funciones al definirlas.
  2. Programa la función `media` que calcula la media aritmética de una lista de números, usando para ello la función `length` que calcula el número de elementos de una lista. ¿Surge algún problema con los tipos? Utiliza la función de conversión `fromIntegral` para resolverlo.
  3. Programa las siguientes funciones, declarando sus tipos:
    - a) `num_digitos x` = número de dígitos del número entero `x`.
    - b) `reduccion x` = resultado del proceso de sumar los dígitos del entero `x`, sumar los dígitos del resultado obtenido, y así sucesivamente hasta obtener un número menor que 10. La reducción de un entero negativo es la de su valor absoluto.
    - c) `comb n m` = número de combinaciones de `n` elementos tomados `m` en `m`
  4. Define la conjunción booleana por ajuste de patrones, pero de tres o cuatro formas diferentes, cambiando el número de ecuaciones, o las combinaciones de patrones `True`, `False`, `x`, ... en cada ecuación, o el orden de ecuaciones, etc. y de manera que al menos haya una versión estricta en el primer argumento y otra estricta en el segundo, pero no en el primero. Para que coexistan todas definiciones en el mismo programa, puedes darles nombres (o usar operadores) diferentes.
  5. Evalúa en el intérprete las siguientes expresiones por ver si tardan poco (digamos centésimas o milésimas de segundos), regular (digamos décimas o segundos) o mucho (digamos toda una vida) en ser evaluadas. Piensa a qué se deben los resultados que obtienes en cada caso.

```
last [1..10^5]
last [1..10^7]
last [1..10^20]
head [1..10^20]
```

```

last [10^20..1]
head (tail [1..10^20])
length [1..10^20]
last (take (10^7) [1..10^20])
head (take (10^7) ([1..100] ++ [1..10^20]))
last (take 100 ([1..10^20] ++ [1..100]))
last (drop 100 ([1..10^20] ++ [1..100]))
head (drop (10^7) ([1..10^20] ++ [1..100]))
[1..10^7]==[1..10^7]
[1..10^20]==[1..10^20]
[1..10^20]==[1..10^20+1]
[1..10^20]==[2..10^20]
head (reverse [1..10^7])
last (reverse [1..10^7])
reverse [1..10^20] == reverse [1..10^20+1]

```

Este ejercicio es para que experimentes. No hace falta que incluyas la solución en el fichero que entregues.