

# PARADIGMAS DE PROGRAMACION

Trabajo Práctico N° 8

Fecha: 31/10/22

## Tema: Programación Funcional. Lenguaje Haskell.

1. Explique qué hace el siguiente código Haskell y de un nombre más adecuado a la función.

```
misterio x
| x <= 0 = 0
| x == 1 = 1
| otherwise = 2*x + misterio (x - 1) - 1
```

2. Implemente en Haskell las siguientes funciones:

- a. **divisiónEntera** que reciba dos números naturales realice la división entera entre dichos números mediante restas sucesivas.
- b. **contarMayores** que dada una lista de números enteros y un número X, cuente cuantos elementos de la lista son mayores que X.
  - i. Realice una versión con *guards*
  - ii. Realice una versión con *pattern matching*
  - iii. Realice una versión con *list comprehension*
- c. **subLista** que reciba una lista y un número natural n y retorne una lista con los primeros n elementos de la lista dada.
- d. **intersección** que reciba dos listas y devuelva una nueva lista con los elementos que tienen en común las listas dadas.
- e. **transformar** que reciba como parámetros una función **f** (de un argumento) y una lista y devuelva como resultado la lista recibida en la que cada uno de sus elementos haya sido transformado con la función **f**.
- f. **tablaDel5** que construya una lista que contenga los números de la tabla del 5 que se encuentran en el intervalo [ 0,50]. Use *List Comprehension*.
- g. **paresOrdenados** que construya una lista con todos los pares ordenados (a,b) posibles teniendo en cuenta que a corresponde a todos los números pares y b a los números impares entre 0 y 5 donde se cumple la condición que  $a < b$ . Use *List Comprehension*.
- h. **verificar** que recibe un predicado p (un predicado es una función que devuelve un valor booleano) y una lista de elementos xs y devuelve True si todos los elementos de la lista satisfacen el predicado, caso contrario retorna False. Realice una versión con guards
- i. **combinar** que reciba una función **f** y dos listas y retorne una nueva lista que resulta de la combinación las listas aplicando la función **f**. La función **f** debe recibir como parámetro un elemento de cada lista a combinar por vez. Utilice la siguiente definición de tipo para su función:  

```
combinarCon :: (a -> b -> c) -> [a] -> [b] -> [c]
```
- j. **filtrarLista** que reciba un predicado y una lista y luego regresa la lista de elementos que satisfacen el predicado. La signatura de la función debería ser:  

```
filtrarLista :: (a -> Bool) -> [a] -> [a]
```

  - i. Realice una versión con *guards*
  - ii. Realice una versión con *list comprehension*

3. Estudie cómo se aplica la función foldl en Haskell y escriba una versión propia.