

# Facultad de Ciencias UNAM

## Lógica Computacional

### Práctica 1

Alejandro Hernández Mora \*

Entrega: Jueves 15 de Febrero de 2018

## 1. Objetivos

Que el alumno tenga una breve introducción a *Haskell* y la programación funcional, así como a utilizar los tipos básicos en *Haskell*, estructuras de control y también la caza de patrones.

## 2. Ejercicios

Programa las siguientes funciones en Haskell:

1. Una función que recibe cuatro enteros y calcula la derivada de la función  $f(x) = ax^2 + bx + c$ . El primer elemento corresponde a  $a$ , el segundo a  $b$ , el tercero a  $c$  y el cuarto a  $x$ .

`deriva:: Int -> Int -> Int -> Int -> Int`

2. Una función que recibe dos números de punto flotante, el primer parámetro corresponde al radio de un cilindro, el segundo a la altura y calcula:

a) El área del cilindro:  $area = 2\pi r(r + h)$

`areaCilindro:: Float -> Float -> Float`

b) El volumen del cilindro:  $volumen = 2\pi r^2 h$

`volumenCilindro:: Float -> Float -> Float`

---

\* `alejandrohroma@ciencias.unam.mx`

3. Una función que recibe tres parámetros, el primero indica la operación que se desea realizar, el segundo un número y el tercero otro número. Las operaciones que se pueden hacer son:
- 's' - Devuelve el segundo parámetro.
  - 't' - Devuelve el tercer parámetro.
  - 'a' - Devuelve la suma de ambos números.
  - 'r' - Devuelve la resta de ambos números.
  - 'p' - Devuelve el producto de ambos números.
  - 'd' - Devuelve la división entera de ambos números.
  - 'e' - Devuelve la potencia (Segundo parámetro elevado al tercer parámetro).

`aplicaOperacion :: Char -> Int -> Int -> Int`

4. Una función que recibe un entero y devuelve la aproximación entera a su raíz cuadrada.

`raizEntera :: Int -> Int`

5. Una función que reciba un entero  $n$  y devuelva la suma de los primeros  $n$  números naturales.

`sumaNat :: Int -> Int`

6. Una función que reciba un entero  $n$  y devuelva la suma de los primeros  $n$  números naturales.

`sumaNat :: Int -> Int`

7. Funciones de orden superior:

- a) Una función que reciba un número  $n$  y devuelva la lista con los primeros  $n$  términos de la sucesión tribonacci que inicia con  $[0, 1, 1]$ . Como requisito extra, debes usar la función `map` para implementar esta función.

`tribonaccies :: Int -> [Int]`

- b) Una función que reciba una lista y elimine los duplicados adyacentes de la lista, dejando una presencia de cada elemento contiguo.

`eliminaDup :: [a] -> [a]`

8. Operaciones con listas.

Redefinir o definir cada una de las siguientes operaciones sobre listas. No está permitido usar funciones de Haskell que resuelvan directamente los ejercicios. **HINT: Explota la caza de patrones de *Haskell*.**

- a) Una función que recibe una lista y devuelve la misma pero en el orden inverso.

`reversa :: [a] -> [a]`

- b) Una función que recibe una lista y devuelve a los elementos de la lista que cumplen el predicado recibido (`filter`).

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

- c) Una función que recibe una lista y devuelve una lista con pares ordenados  $(k, x)$ , donde  $k$  es el máximo número de apariciones consecutivas del elemento  $x$ .

```
apariciones :: [a] -> [(Int,a)]
```

```
sumaNat :: Int -> Int
```

9. Reescribir las siguientes listas como listas por comprensión.

```
[0, 1, 3, 7, 15, 31, 63]
```

```
[(3, 4), (7, 8), (11, 12), (15, 16), ...]
```

Deberás respetar todas las definiciones de las funciones, es decir que esta prohibido cambiar el tipo que tengan, ya sea el de algún parámetro o el del resultado.

### 3. Requerimientos

Deberás entregar tu práctica al correo **luismanuel@ciencias.unam.mx** (y sólo a ese correo) antes de las 23:59 del día jueves 15 de febrero de 2018, tal y como lo indican los lineamientos de entrega, de lo contrario la práctica podría no ser calificada.

Deberás enviar por correo un archivo llamado **Practica1.hs**. El orden en el que definan las funciones en el archivo **Practica1.hs** debe ser el orden especificado en este PDF, de lo contrario se bajarán puntos.

¡Que tengas éxito en tu primera práctica!.