

# Implementación de Modelos Computacionales TC2037-15

## Extra 01 – Programación Funcional EX

*En esta tarea, practicarán con Haskell para implementar múltiples funciones. Por favor consideren que el objetivo de esta tarea es permitirles practicar e identificar fuerzas y áreas de oportunidad con respecto a su capacidad. Por lo mismo, se sugiere que no utilicen funciones ya implementadas. Para facilitar cuestiones de diseño, consideren que reciben solamente argumentos del tipo de dato adecuado.*

### 1. Challenge IV (+1)

Genera una función en Haskell que, dado un árbol binario de búsqueda y un nodo, indique la profundidad de dicho nodo. Considera que la raíz de un árbol tiene profundidad 0. Se recomienda que para representar un árbol, utilices el formato siguiente `[root [left] [right]]`. El nodo `root` tiene como hijo izquierdo a la lista `left` (el subárbol izquierdo), y como hijo derecho al subárbol `right`.

### 2. Challenge V (+1)

Genera una función en Haskell que, dado un grafo, indique el mayor de sus grados (es decir, el número de conexiones del nodo más conectado). Puedes usar cualquier representación de grafo que prefieras.

- Una lista de adyacencia: `[[a, b, c], [b, a, d], [c, a, d], [d, b, c]]`
- Una matriz de adyacencia:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

- Una tupla de vértices y ejes: `([a, b, c, d], [[a,b], [a, c], [b, a], ...])`

## Entregables



Preparen un archivo `.hs` que contenga las funciones requeridas y súbanlo a Canvas en el apartado correspondiente. Por favor, no suban código fuente en otro formato no editable.



En esta actividad me comprometo conmigo y mi equipo a asumir un rol activo honesto y responsable, basado en la confianza y la justicia y a no servirme de medios no autorizados o ilícitos para realizarla, siguiendo el Código de Ética del Tecnológico de Monterrey.