



Universidad del Istmo de Guatemala  
Facultad de Ingenieria  
Ing. en Sistemas  
Informatica 1  
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

---

## Hoja de trabajo #8

Fecha de entrega: 11 de Octubre, 2018 - 11:59pm

---

*Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Hoja de trabajo 8". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante LaTeX.*

**Nota:** En este deber se omitira la ubicación exacta del compilador de elm, y solo se escribira elm. Por ejemplo, en vez de escribir:

```
> node_modules\elm repl
```

Se escribira:

```
> elm repl
```

Adicionalmente, asegurarse que las funciones y modulos que sean declarados en su deber correspondan exactamente a los nombres escritos en dicho deber ya que se utilizaran pruebas automatizadas para calificar.

### Ejercicio #1 (30%)

Defina la función `zipWith` :  $(a \rightarrow b \rightarrow c) \rightarrow \text{List } a \rightarrow \text{List } b \rightarrow \text{List } c$  la cual debe aceptar una función como parametro y utilizar dicha función para combinar los valores de las dos listas pasadas como parametro y producir una lista con la combinación de los valores.

### Ejercicio #2 (35%)

Defina la función `groupBy` :  $(a \rightarrow \text{Bool}) \rightarrow \text{List } a \rightarrow (\text{List } a, \text{List } a)$  la cual acepta una función que actua de clasificador colocando en la la primera lista del resultado todos los elementos para los cuales dicha función es `True` y en la segunda los elementos para los cuales la función retorna `False`. Ejemplo:

- `groupBy`  $(\lambda x \rightarrow x < 3)$  `[1, 2, 3, 4]`  
– `([1, 2], [3, 4])`

## Ejercicio #3 (35%)

Elm trae integrado un tipo llamado **Maybe** el cual tiene dos constructores: `'Just : a → Maybe a'` y `'Nothing : Maybe a'`. Este tipo se utiliza para representar valores opcionales o condiciones erróneas.

Su tarea es definir la función `bind : Maybe a → (a → Maybe b) → Maybe b` la cual toma como primer parametro un valor de tipo `Maybe a`, el cual es un valor opcional, una función que transforma el valor `a` a un `Maybe b` y produce un `Maybe b`. Si el primer parametro es `Nothing`, la función simplemente debe retornar `nothing`. Ejemplos de su utilización:

- `bind (Just 5) (λi → Just (i + 3))`  
– `Just 8`
- `bind Nothing (λi → Just (i + 3))`  
– `Nothing`