



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

---

## Hoja de trabajo #5

Fecha de entrega: 6 de Septiembre, 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 5". 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 (20%)

Crear un proyecto utilizando elm entro de la carpeta "Hoja de trabajo 5". Para ello debe ejecutar el comando:

```
> elm init
```

Y responder "Y" a la pregunta. Puede ver el contenido de la carpeta "ejemplo" para observar la estructura de un proyecto de elm. Luego, dentro de la carpeta "src" que ha sido creada en el proyecto, colocar un archivo llamado "Main.elm". Por ultimo, crear un modulo llamado "Main" en dicho archivo agregando el texto:

```
module Main exposing ()
```

Al principio del archivo.

## Ejercicio #2 (20%)

Dentro del modulo “Main”, definir una función `esPrimo` : `Int` → `Bool` la cual acepta un entero y retorna `True` si el numero es primo o `False` de lo contrario. No es necesario hacer logica especial para numeros negativos. Luego de definir la función, asegurarse que:

1. Que la función tenga una **firma**, ver el ejemplo en “ejemplo/src/Main.elm”, para ver como se le coloca la firma a una función.
2. Que el modulo “Main” **exponga** la función, para ello debe cambiar “`module Main exposing ()`” a “`module Main exposing (esPrimo)`”
3. Puede probar que su función esta implementada correctamete en la terminal interactiva. Para ello:
  - (a) Abra la la terminal interactiva con `elm repl`
  - (b) Ejecute `import Main` en la terminal interactiva
  - (c) Llame a la función con un valor: `Main.esPrimo 42`

De este momento en adelante, toda función que sea declarada debe tener una **firma** y el modulo debe **exponer** dichas funciones.

## Ejerciico #3 (20%)

La sequencia de Fibonacci se define asi:

$$\text{fibonacci}(n) = \begin{cases} 0 & \text{if } x \equiv 0 \\ 1 & \text{if } x \equiv 1 \\ \text{fibonacci}(n-1) + \text{fibonacci}(n-2) & \text{de lo contrario} \end{cases}$$

## Ejercicio #4 (20%)

Definir la función `primos` : `Int` → `List Int` la cual retorna una lista de todos los primos menores o iguales a su primer parametro.

## Ejercicio #5 (20%)

Definir la función `nPrimos` : `Int` → `List Int` la cual retorna los pirmeros  $n$  primos donde  $n$  es el primer parametro de la función.