

# Simulacro de Primer Parcial (Nivel 2)

## Algoritmos y Estructuras de Datos I

Práctica de Repaso - MDO

Noviembre 2025

### Programación Funcional (Haskell)

#### Pregunta 1 [3 puntos]

Dadas las siguientes definiciones en Haskell:

```
-- Funci n que eval a su primer argumento para decidir
g :: Int -> Int -> Int
g 0 y = 5
g x y = x

-- Loop infinito
loop :: Int
loop = loop + 1

-- Lista infinita
nums :: [Int]
nums = 1 : nums
```

Evaluar paso a paso las siguientes expresiones utilizando **\*\*Orden Normal\*\***. Indicar si convergen a un valor o divergen (se cuelgan/dan error).

- a) `g 0 loop`
- b) `g 10 loop`
- c) `g loop 10`
- d) `head (0 : nums)`

*(Pista: Recuerde que el Orden Normal en Haskell se vuelve aplicativo cuando se debe evaluar un patrón (pattern matching)).*

### Pregunta 2 [2 puntos]

Se desea escribir una función `divisores` utilizando **listas por comprensión**. La función debe recibir un número entero `n` y devolver una lista con todos sus divisores propios (es decir, todos sus divisores excepto `n` mismo).

**Ejemplo:**

```
> divisores 12
[1, 2, 3, 4, 6]
```

### Pregunta 3 [2 puntos]

Definir una función recursiva `reversar` que tome una lista y devuelva la misma lista en orden inverso. (*No puede usar un fold ni la función `reverse` predefinida. Pista: el operador `(++)`*)

**Completar el siguiente esquema:**

```
reversar :: [a] -> [a]
reversar []      = ...
reversar (x:xs) = ...
```

### Pregunta 4 [3 puntos]

Dada la función `map` de Haskell (que aplica una función a cada elemento de una lista), se pide re-implementarla (con el nombre `mapear`) utilizando `foldr`.

**Completar la definición:**

```
mapear :: (a -> b) -> [a] -> [b]
mapear f xs = foldr ... ... xs
```

(Pista: Piense: ¿cuál es el caso base (lista vacía) y qué operación combina el elemento `x` con el acumulador `acc`?)

## Lógica y Verificación (Imperativo)

### Pregunta 5 [2 puntos]

Indicar si las siguientes ternas de Hoare son Válidas o Falsas. Justificar calculando la WP y verificando la implicación  $P \implies WP$ .

1.  $\{x > y\} \quad x := x - y \quad \{x > 0\}$
2.  $\{True\} \quad x := 5; \quad y := x + 1 \quad \{y == 5\}$

**Pregunta 6 [3 puntos]**

Calcular la **\*\*Precondición Más Débil (Weakest Precondition - WP)\*\*** del siguiente programa para que satisfaga la postcondición dada. Simplificar la expresión lógica resultante.

**Programa:**

```
if (x > 10) then
  if (x > 20) then
    y := 1
  else
    y := 2
else
  y := 3
```

**Postcondición:**  $\{y \geq 2\}$