

Facultad de Ciencias - UNAM
Estructuras Discretas 2026-1
Práctica 1: Introducción a Haskell, relaciones y números naturales

Favio Ezequiel Miranda Perea
Patricio Ordoñez Blanco
Eduardo Vargas Pérez

5 de Septiembre de 2025
Fecha de entrega: 19 de Septiembre de 2025 hasta las 23:59

Desarrollo de la práctica

Funciones

Realiza la implementación de cada una de las siguientes funciones. No se necesita recursión para estas funciones.

1. `valorAbs :: Int -> Int`

Dado un número entero, esta función debe calcular el valor absoluto de ese número haciendo uso de *if_then_else*. **[0.5 Puntos]**

Ejemplos

```
ghci > valorAbs 5 ▷ 5  
ghci > valorAbs (-10) ▷ 10
```

2. `esDivisor :: Int -> Int -> Bool`

Dados dos números enteros, esta función debe devolver *True* si el primer número es divisor del segundo; *False* en otro caso. **[0.5 Puntos]**

Ejemplos

```
ghci > esDivisor 5 10 ▷ True  
ghci > esDivisor 3 5 ▷ False
```

3. `cuadratica :: Float -> Float -> Float -> Float -> Float`

La función recibe 4 parámetros a, b, c, v y debe calcular la evaluación de la ecuación $ax^2 + bx + c$ en v . **[0.5 Puntos]**

Ejemplos

```
ghci > cuadratica 1 0 0 2 ▷ 4.0  
ghci > cuadratica 0.5 1 2 2.5 ▷ 7.625
```

4. `sumaFracciones :: (Int, Int) -> (Int, Int) -> (Int, Int)`

La función recibe dos tuplas que representa a las fracciones $\frac{a}{b}$ y $\frac{b}{c}$. Se debe devolver una tupla con el resultado de sumar las fracciones que recibe como parámetros. No es necesario simplificar el resultado, pero sí hacer la diferencia cuando las fracciones tienen el mismo denominador y cuando no. **[0.5 Puntos]**

Ejemplos

```
ghci > sumaFracciones (1,4) (2,4) ▷ (3,4)
ghci > sumaFracciones (1,2) (1,4) ▷ (6,8)
```

5. `comparador :: Float -> Float -> Int`

La función debe recibir dos números n, m . Se debe devolver 0 si $n = m$, 1 si $n > m$ y -1 si $m > n$. **[0.5 Puntos]**

Ejemplos

```
ghci > comparador 5 5 ▷ 0
ghci > comparador 5.5 5 ▷ 1
ghci > comparador 5.5 9.2 ▷ -1
```

6. `puntoMedio :: (Float,Float) -> (Float,Float) -> (Float,Float)`

La función debe recibir dos puntos del plano cartesiano y devolver el punto medio entre ellos. **[0.5 Puntos]**

Ejemplo

```
ghci > puntoMedio (9.2,7) (3, -13.5) ▷ (6.1,-3.25)
```

Relaciones

Sean A, B dos conjuntos tales que

$$A = B = \{1, 2, 3, \dots, 30\}$$

Considera los ejemplos de relaciones vistas en la ayudantía de laboratorio e implementa las siguientes relaciones de A en B :

1. `relacionDivisor :: Rel Int Int`

En esta relación R_1 , tenemos que aR_1b si a y b tienen la misma paridad y a es divisor de b .

[1 Punto]

Ejemplos

$$(2, 6) \in R_1 \quad \text{pero} \quad (3, 6) \notin R_1$$

2. `relacionSumaEspecial :: Rel Int Int`

En esta relación R_2 , tenemos que aR_2b si $a + b$ es múltiplo de 5 y $a < b$. **[1 Punto]**

Ejemplos

$$(2, 8) \in R_2 \quad \text{pero} \quad (8, 2) \notin R_2$$

3. `relacionCongruentesModuloN :: Int -> Rel Int Int`

En esta relación R_3 , se debe recibir un entero n y tenemos que aR_3b si $a \% n = b \% n$ con $a \neq b$.

[1 Punto]

Ejemplos

Suponiendo $n = 5$

$$(5, 10) \in R_3 \quad \text{pero} \quad (5, 5) \notin R_3$$

Números naturales

Considera el tipo de dato `Natural` que se definió en la ayudantía de laboratorio e implementa las siguientes funciones usando recursión:

1. `esPar :: Natural -> Bool`

La función debe recibir un número natural y determinar si es par o no. **[1 Punto]**

Ejemplos

```
ghci > esPar (Suc (Suc (Suc Cero))) > False
ghci > esPar (Suc (Suc (Suc (Suc Cero)))) > True
```

2. `iguales :: Natural -> Natural -> Bool`

La función debe recibir dos números naturales y determinar si estos números son iguales o no.

[1 Punto]

Ejemplos

```
ghci > iguales (Suc (Suc (Suc Cero))) (Suc (Suc (Suc Cero))) > True
ghci > iguales (Suc (Suc Cero)) (Suc (Suc (Suc Cero))) > False
```

3. `maximo :: Natural -> Natural -> Natural`

La función debe recibir dos números naturales y devolver el que sea mayor. **[1 Punto]**

Ejemplo

```
ghci > maximo (Suc Cero) (Suc (Suc Cero)) > (Suc (Suc Cero))
```

4. `potencia :: Natural -> Natural -> Natural`

La función debe recibir dos números naturales n, m y devolver el resultado de n^m . **[1 Punto]**

Ejemplo

```
ghci > potencia (Suc (Suc Cero)) (Suc (Suc Cero))
> (Suc (Suc (Suc (Suc Cero))))
```

Limitaciones

Recuerden que toda función auxiliar que necesiten (con excepción de las funciones *div* y *mod*) tiene que ser implementada por ustedes.

¡Buena suerte a todos! ☺☺☺