

¿Clases en Haskell?

Eduardo Vargas Perez

Facultad de Ciencias

12 de septiembre de 2025

Clases de tipo en Haskell

Haskell hace uso de clases de tipo como una manera de asociar ciertas funciones a algún tipo en particular.

Si lo quieren trasladar a su curso de ICC, se parece un poco a las interfaces en Java. Es una especie de garantía de que el tipo tendrá disponibles ciertas funciones y operaciones.

Ejemplos pasados

¿Recuerdan la línea deriving?

```
data Natural = Cero | Suc Natural deriving (Show,Eq)
data CadenaBinaria = Vacia
    | O CadenaBinaria
    | I CadenaBinaria
    deriving (Show, Eq)
```

Algunas clases de tipo que posiblemente usemos

Aunque existen más o incluso pueden definir sus propias clases de tipo, sólo hablaremos de las siguientes:

- Eq
- Ord
- Show
- Read
- Num
- Enum

Eq

Nos garantiza que vamos a poder comparar si dos cosas de un mismo tipo son iguales o distintas.

```
class Eq a where
    (==) :: a -> a -> Bool
    (/=) :: a -> a -> Bool
```

Ord

Nos garantiza que vamos a poder comparar con respecto a un orden.

```
class Eq a => Ord a where
  (<), (<=), (>), (>=) :: a -> a -> Bool
  compare :: a -> a -> Ordering  -- LT, EQ, GT
```

Show

Sirve para convertir los valores a texto (cadenas String).

```
class Show a where  
  show :: a -> String
```

Read

Para hacer lo contrario a Show. A partir de una cadena, devolver el tipo.

```
class Read a where  
  read :: String -> a
```


Num

Permite usar las operaciones aritméticas.

```
class Num a where  
    (+), (-), (*) :: a -> a -> a  
    negate :: a -> a
```

Enum

Sirve para tipos que se pueden enumerar secuencialmente. Permite usar rangos (`[a..b]`) y sucesores (`succ`, `pred`)

```
class Enum a where
  succ, pred :: a -> a
  toEnum :: Int -> a
  fromEnum :: a -> Int
```

Implementaciones propias

Es posible hacer nuestras implementaciones propias.

```
instance Eq Shape where
    s1 == s2 = area s1 == area s2

instance Ord Shape where
    compare s1 s2 = compare (perimetro s1) (perimetro s2)
```

En el ejemplo de las figuras, estamos diciendo que dos figuras son iguales si tienen la misma área. Sin embargo, las ordenamos conforme al perímetro.