

What is Functor and how to apply it?

Santiago Caballero Manzaneda

Funtores en Haskell

- ¿Qué es un functor?

- Un functor es un concepto en Haskell que nos permite aplicar funciones a los valores dentro de una estructura de datos mientras mantenemos la estructura intacta.

- ¿Por qué son útiles?

- Los funtores nos permiten realizar operaciones sobre estructuras de datos de manera abstracta y general, lo que promueve el código más limpio y modular.

- **Sintaxis de Functor en Haskell:**

```
haskellCopy code
class Functor f where
    fmap :: (a -> b) -> f a -> f b
```

Ejemplo de Functor en Acción

- **Definición de una estructura de datos:**

```
haskellCopy code
data Lista a = Vacia | Cons a (Lista a) deriving (Show)
```

- **Instancia de Functor para Lista:**

```
haskellCopy code
instance Functor Lista where
    fmap _ Vacia          = Vacia
    fmap f (Cons x xs)    = Cons (f x) (fmap f xs)
```

- **Uso de Functor en una lista:**

```
haskellCopy code
incrementar :: Int -> Int
incrementar x = x + 1

listaOriginal = Cons 1 (Cons 2 (Cons 3 Vacia))
listaModificada = fmap incrementar listaOriginal
```

Slide 3: Beneficios y Conclusiones

- **Beneficios de usar Functores:**

- Abstracción sobre estructuras de datos.
- Código más limpio y modular.
- Promueve la programación funcional.

- **Conclusiones:**

- Los funtores son una herramienta poderosa en Haskell para operar sobre estructuras de datos de manera abstracta.
- Son parte integral de la programación funcional en Haskell y ayudan a escribir código más expresivo y mantenible.