

Paradigmas de Lenguajes de la Programación

Manuel Mena

25 de marzo de 2019

Índice

1. Practica 0	2
1.1.	2

1. Practica 0

1.1.

`null :: Foldable t => t a -> Bool`

Chequea si la estructura es nula

`head :: [a] -> a`

Devuelve el primer elemento de la lista, mientras sea no vacía

`tail :: [a] -> a`

Devuelve el último elemento de la lista, mientras sea no vacía

`init :: [a] -> [a]`

Devuelve todos los elementos de la lista excepto el último, mientras sea no vacía

`last :: [a] -> [a]`

Devuelve todos los elementos de la lista excepto el primero, mientras sea no vacía

`take :: Int -> [a] -> [a]`

Devuelve los primeros n elementos de la lista

`drop :: Int -> [a] -> [a]`

Devuelve una lista sin sus primeros n elementos

`(++) :: [a] -> [a] -> [a]`

Concatena ambas listas. Si la primera lista es infinita, devuelve la primera

`concat :: Foldable t => t [a] -> [a]`

The concatenation of all the elements of a container of lists

`(!!) :: [a] -> Int -> a`

Devuelve el n-ésimo elemento

`elem :: Eq a => a -> t a -> Bool`

Devuelve si el elemento ocurre en la estructura