

Ejercicio 1

Dar el tipo y describir el comportamiento de las siguientes funciones del módulo Prelude de Haskell:

`null`, `head`, `tail`, `init`, `last`, `take`, `drop`, `(++)`, `concat`, `(!!)`, `elem`

`null`:

- Tipo: `[a] -> Bool`

- Descripción: Devuelve `True` si la lista está vacía, `False` en caso contrario.

`head`:

- Tipo: `[a] -> a`

- Descripción: Devuelve el primer elemento de la lista.

`tail`:

- Tipo: `[a] -> [a]`

- Descripción: Devuelve la lista sin su primer elemento.

`init`:

- Tipo: `[a] -> [a]`

- Descripción: Devuelve la lista sin su último elemento.

`last`:

- Tipo: `[a] -> a`

- Descripción: Devuelve el último elemento de la lista.

`take`:

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

- Descripción: Toma los primeros `n` elementos de una lista y devuelve una nueva lista.

`drop`:

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

- Descripción: Elimina los primeros `n` elementos de una lista y devuelve el resto.

`(++)` (Concatenación de listas):

- Tipo: `[a] -> [a] -> [a]`

- Descripción: Concatena dos listas.

`concat`:

- Tipo: `[[a]] -> [a]`

- Descripción: Concatena una lista de listas en una sola lista.

`(!!)` (Indexación de lista):

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

- Descripción: Devuelve el elemento en la posición especificada de la lista.

`elem`:

- Tipo: `Eq a => a -> [a] -> Bool`

- Descripción: Devuelve `True` si un elemento está presente en una lista, `False` en caso contrario. La función toma un elemento y una lista, y verifica si el elemento está en la lista.