

## Clase 2 - Esquemas de recursión y tipos de datos

```
map :: (a -> b) -> [a] -> [b]
filter :: (a -> Bool) -> [a] -> [a]

()

map filter :: [a] -> Bool -> [[a] -> [a]]
```

### Recursión estructural

g no se puede llamar a si misma, ni a la cola de la lista salvo en el caso que g se llame a la cola de la lista

Es **recursión estructural** si:

- Hay un caso base fijo de un valor z independiente de g
- El caso recursivo no puede usar los parámetros g ni xs,

salvo en la expresión (g xs)

### foldr, definición

```
foldr :: (a -> b -> b) -> b -> [a] -> b
foldr f [] = z
foldr f z (x:xs) = f x : (foldr f z xs)
```

### Recursión primitiva

Sea  $g :: [a] \rightarrow b$  tal que:  $g [] = \text{< caso base >}$   $g (x:xs) = \text{< caso recursivo >}$

$$\sum_x f$$