

Programación Funcional



Universidad Autónoma
de Entre Ríos

```
qsort1 :: Ord a => [a] -> [a]
qsort1 []      = []
qsort1 (p:xs) = qsort1 lesser ++ [p] ++ qsort1
greater
  where
    lesser = [ y | y <- xs, y < p ]
    greater = [ y | y <- xs, y >= p ]
```

Programación Funcional

- Cuando uno ve este algoritmo en Haskell nota claramente que programar de forma declarativa tiene sus recompensas. Este algoritmo (para el que no se dio cuenta) es el quicksort.
- El algoritmo nos indica (usando pattern matching) que si viene una lista vacía, retorne una lista vacía. Si no es el caso, retorne una lista conformada por los menores del primer elemento ordenados con quicksort, el primer elemento y los mayores al primer elemento también ordenados con quicksort.
- Entre otros detalles podemos ver que se utiliza pattern matching, recursividad y listas por compresión.
- Es muy claro lo que hace tanto que no es necesario documentar en lo más mínimo. No es necesario ningún comentario.
- Comparto esto para que comulguen con mi idea que todo se puede mejorar y vale la pena seguir estudiando para hacer código bonito