

PRÁCTICO NÚMERO 2 – PROGRAMACIÓN DECLARATIVA

VENTURA GINO

EJERCICIO 1:

Dada una lista de enteros, devuelve la lista de los cuadrados de los elementos positivos, en el mismo orden (usar la función anterior).

CÓDIGO:

```
filterPositivos [] = []  
  
filterPositivos (x:xs) | x>0 = x*x : filterPositivos xs  
                        | otherwise = filterPositivos xs  
  
Prelude> :load "Practico2-Ejercicio1.hs"  
[1 of 1] Compiling Main          ( Practico2-Ejercicio1.hs, interpreted )  
Ok, one module loaded.  
*Main> filterPositivos [1, -5, 2, -6, 3, -7]  
[1,4,9]
```

EJERCICIO 2)

Definir la función esPositivo tal que esPositivo se verifica si x es positivo.

Por ejemplo: esPositivo 5 -> True

CÓDIGO:

```
comprobarNumero :: Int->Bool  
  
comprobarNumero x = if x>0 then True else False  
  
Prelude> :load "Practico2-Ejercicio2.hs"  
[1 of 1] Compiling Main          ( Practico2-Ejercicio2.hs, interpreted )  
Ok, one module loaded.  
*Main> comprobarNumero 5  
True  
*Main> comprobarNumero (-9)  
False
```