

# INTRODUCCIÓN

## FUNCIONES PREDEFINIDAS EN Haskell

- $x+y$  es la suma de  $x$  e  $y$ .
- $x-y$  es la resta de  $x$  e  $y$ .
- $x*y$  es el producto de  $x$  e  $y$ .
- $x/y$  es el cociente de  $x$  entre  $y$ .
- $x^n$  es  $x$  elevado a  $n$ , con  $n$  un número natural.
- $x^{\wedge}n$  es  $x$  elevado a  $n$ , con  $n$  un número entero.
- $x^{**}y$  es  $x$  elevado a  $y$ .
- $x == y$  se verifica si  $x$  es igual a  $y$ .
- $x /= y$  se verifica si  $x$  es distinto de  $y$ .
- $x < y$  se verifica si  $x$  es menor que  $y$ .
- $x <= y$  se verifica si  $x$  es menor o igual que  $y$ .
- $x > y$  se verifica si  $x$  es mayor que  $y$ .
- $x >= y$  se verifica si  $x$  es mayor o igual que  $y$ .
- $x \&\& y$  es la conjunción de  $x$  e  $y$ .
- $x || y$  es la disyunción de  $x$  e  $y$ .
- $x:ys$  es la lista obtenida añadiendo  $x$  al principio de  $ys$ .
- $xs ++ ys$  es la concatenación de  $xs$  e  $ys$ .
- $xs !! n$  es el elemento  $n$ -ésimo de  $xs$ .
- $f . g$  es la composición de  $f$  y  $g$ .
- $\text{abs } x$  es el valor absoluto de  $x$ .
- $\text{and } xs$  es la conjunción de la lista de booleanos  $xs$ .
- $\text{ceiling } x$  es el menor entero no menor que  $x$ .
- $\text{concat } xss$  es la concatenación de la lista de listas  $xss$ .

- `concatMap f xs` aplica la función `f` a los elementos de `xs` y concatena los resultados.
- `curry f` es la versión curryficada de la función `f`.
- `div x y` es la división entera de `x` entre `y`.
- `divMod x y` es el par formado por el cociente de `x` entre `y` y su módulo.
- `drop n xs` borra los `n` primeros elementos de `xs`.
- `elem x ys` se verifica si `x` pertenece a `ys`.
- `filter p xs` es la lista de elementos de la lista `xs` que verifican el predicado `p`.
- `floor x` es el mayor entero no mayor que `x`.
- `fst p` es el primer elemento del par `p`.
- `gcd x y` es el máximo común divisor de `x` e `y`.
- `head xs` es el primer elemento de la lista `xs`.
- `init xs` es la lista obtenida eliminando el último elemento de `xs`.
- `last xs` es el último elemento de la lista `xs`.
- `lcm x y` es el mínimo común múltiplo de `x` e `y`.
- `length xs` es el número de elementos de la lista `xs`.
- `map f xs` es la lista obtenida aplicado `f` a cada elemento de `xs`.
- `max x y` es el máximo de `x` e `y`.
- `maximum xs` es el máximo elemento de la lista `xs`.
- `min x y` es el mínimo de `x` e `y`.
- `minimum xs` es el mínimo elemento de la lista `xs`.
- `mod x y` es el módulo de `x` respecto de `y`.
- `not x` es la negación lógica del booleano `x`.
- `notElem x ys` se verifica si `x` no pertenece a `ys`.
- `null xs` se verifica si `xs` es la lista vacía.
- `odd x` se verifica si `x` es impar.

- `or xs` es la disyunción de la lista de booleanos `xs`.
- `product xs` es el producto de la lista de números `xs`.
- `quot x y` es el cociente entero de `x` entre `y` truncado hacia cero.
- `rem x y` es el resto de `x` entre `y`.
- `x` es la lista infinita `[x, x, x, ...]`.
- `replicate n x` es la lista formada por `n` veces el elemento `x`.
- `reverse xs` es la inversa de la lista `xs`.
- `round x` es el redondeo de `x` al entero más cercano.
- `signum x` es 1 si `x` es positivo, 0 si `x` es cero y -1 si `x` es negativo.
- `snd p` es el segundo elemento del par `p`.
- `sqrt x` es la raíz cuadrada de `x`.
- `sum xs` es la suma de la lista numérica `xs`.
- `tail xs` es la lista obtenida eliminando el primer elemento de `xs`.
- `take n xs` es la lista de los `n` primeros elementos de `xs`.
- `truncate x` es la parte entera de `x`.
- `zip xs ys` es la lista de pares formado por los correspondientes elementos de `xs` e `ys`.
- `xs ys zs` es la lista de pares formado por los correspondientes elementos de `xs`, `ys` y `zs`.

...