

I. Definición y Tipo Básico

Definir las siguientes funciones y determinar su tipo más general:

1. **triple**: que toma un número y devuelve su triple.
2. **second**: que toma un par ordenado, y devuelve su segunda componente.
3. **constA**: que toma cualquier valor x y cualquier valor y , y devuelve x .
4. **compose**: que toma dos funciones f y g y devuelve el resultado de aplicar f al resultado de aplicar g a un valor x (es decir, $f(g\ x)$).

II. Ejemplos de Tipos

Dar al menos dos ejemplos de funciones que tengan cada uno de los siguientes tipos:

1. $\text{Int} \rightarrow \text{Char}$
2. $a \rightarrow \text{Int}$
3. $(\text{Int}, \text{Int}) \rightarrow (\text{Int}, \text{Int})$
4. $((a \rightarrow b), a) \rightarrow b$

III. Reescribir Sin let, where o if

Reescribir cada una de las siguientes definiciones sin usar las palabras clave `let`, `where` o `if`, utilizando `pattern matching` o `guardas`:

1. $f\ x = \text{if } x > 0 \text{ then } x \text{ else } 0$
2. $\text{esCero } x = \text{if } x == 0 \text{ then True else False}$
3. $\text{invertirPar } (x,y) = \text{let } z = (y,x) \text{ in } z$
4. $\text{esUno } x = \text{let } y = 1 \text{ in if } x == y \text{ then True else False}$

IV. Inferencia y Composición de Tipos

1. Suponiendo que $f :: b \rightarrow c$ y $g :: a \rightarrow b$, determinar el tipo de h si $h\ x = f\ (g\ x)$.
2. Si $f :: \text{Int} \rightarrow \text{Int} \rightarrow \text{Bool}$ y $g :: \text{Int} \rightarrow \text{Int}$, determinar el tipo de h si $h\ x\ y = f\ (g\ x)\ y$.
3. Sean $f :: (a, b) \rightarrow a$ y $g :: c \rightarrow (d, e)$. Determinar el tipo de h si $h\ z = f\ (g\ z)$.
4. Si $f :: a \rightarrow b$ y $g :: b \rightarrow c$, determinar cuál de las siguientes definiciones de h es equivalente a $h = g \circ f$:
 - $h\ x = g\ (f\ x)$
 - $h = g \circ f$

V. Funciones con Guardas Complejas

1. **diaValido**: Definir una función que determine si un número dado es un día válido del mes (entre 1 y 31).
2. **esTrimestre**: Definir una función que tome un número entero y devuelva `True` si ese número corresponde al número de un mes del primer trimestre del año (enero, febrero, o marzo).
3. **aprobado**: Definir una función que tome la nota de un examen (entero de 0 a 10) y devuelva: "Sobresaliente" si la nota es mayor o igual a 9, "Notable" si es mayor o igual a 7, "Aprobado" si es mayor o igual a 4, y "Reprobado" en caso contrario.
4. **validaPassword**: Definir una función que tome una longitud y una cadena, y devuelva `True` si la cadena tiene exactamente esa longitud.

VI. Recursión Explícita

Sin usar funciones predefinidas de alto nivel (map, filter, fold, etc.), defina recursivamente las siguientes funciones:

1. **longitud:** que calcula el número de elementos de una lista.
2. **concatena:** que toma dos listas y devuelve una nueva lista que es la concatenación de ambas.
3. **contarMayores:** que toma un número n y una lista de números, y devuelve la cantidad de elementos en la lista que son mayores que n .
4. **maximo:** que devuelve el elemento máximo de una lista de números.

VII. Listas por Comprensión

Definir las siguientes funciones usando listas por comprensión:

1. **multiplos:** que dado un número n y un límite m , devuelve la lista de los múltiplos de n que son menores o iguales a m ($n, 2n, 3n, \dots$).
2. **todosIguales:** que dada una lista de valores, devuelve una lista de pares donde cada par está formado por el elemento y `True` si todos los elementos de la lista son iguales a ese elemento.
3. **pitagoricas:** que devuelve la lista de todas las ternas (a, b, c) con $0 < a, b, c \leq 20$ que cumplen la relación pitagórica $a^2 + b^2 = c^2$.
4. **paresDeLaSuma:** que dados un límite n y un objetivo t , devuelve la lista de todos los pares (x, y) con $0 < x, y \leq n$ cuya suma sea igual al objetivo t ($x + y = t$).