

# PROGRAMACIÓN FUNCIONAL

## Trabajo Práctico Nro. 2

**Temas:** Expresiones y valores. Tipos. Notación Lambda.

### Bibliografía relacionada:

- Simon Thompson. The craft of Functional Programming. Addison Wesley, 1996. Cap. 2.
- L.C. Paulson. ML for the working programmer. Cambridge University Press, 1996. Cap. 2.

1. De ser posible, dar dos ejemplos de expresiones que tengan cada uno de los siguientes tipos:

- |                               |                               |
|-------------------------------|-------------------------------|
| <i>a)</i> Bool                | <i>e)</i> (Int -> Int) -> Int |
| <i>b)</i> (Int, Int)          | <i>f)</i> (Bool -> Bool, Int) |
| <i>c)</i> Char -> Int         | <i>g)</i> a -> Bool           |
| <i>d)</i> (Int, Char) -> Bool | <i>h)</i> c -> c              |

2. ¿Qué ventajas tiene el uso de tipos en un lenguaje de programación?
3. Dar los tipos de las funciones de la Práctica 1, ejercicio 1.
4. Dar los tipos de las siguientes funciones y verificar el resultado en Hugs:

- a)* first (x,y) = x
- b)* second (x,y) = y
- c)* const x y = x
- d)* compose f g = (\x -> f (g x))
- e)* apply f x = f x
- f)* subst f g x = f x (g x)
- g)* pairFunc (f1,f2) x y = (f1 (f2 x), f2 (f1 y))

5. Definir tipado estático.
6. Para las siguientes expresiones, indicar si están bien formadas. De ser así, dar su valor o en caso contrario, especificar si el error es sintáctico, de tipos, o de alguna otra clase:

- a) `if (seven 'a' < 1) then 5 else power4 2`
- b) `if False then True`
- c) `a := 4`
- d) `(1 < x && x < 100) || x == True || x == False`
- e) `False == (1 < 3)`
- f) `(1 < x < 2)`

7. ¿Qué entiende por polimorfismo?

8. Defina los siguientes tipos, con las operaciones especificadas.

- `ColorPrimario`
- `ColorSecundario` y la operación `mezclar`, la cual combina dos colores primarios (nunca se mezcla el mismo color primario).
- `Punto`, para dos y tres coordenadas, con las operaciones `modulo`, `distanciaA`, `xcoord`, `ycoord` y `suma`.
- `Figura`, para representar círculos y rectángulos en el plano. Además, definir las operaciones `area`, `perimetro` y `mover`.
- `Figura3D` que puede ser un cubo, un cilindro o una esfera. Definir las operaciones `area` y `volumen`.
- `Persona`, el cual tiene `Nombre`, `Edad`, `Documento`, `Dirección` y `Teléfono`. Distinguir entre personas que trabajan y no, sabiendo que las que trabajan tienen además una `dirección` y un `teléfono` laboral.

9. Dar al menos dos ejemplos de expresiones que tengan los siguientes tipos:

- a) `a`
- b) `Int -> a`
- c) `a -> b`
- d) Dar una expresión de tipo `c -> c`, distinta de la identidad.

10. (★) ¿Cuál es el tipo de la función `tom` en Haskell?

`tom x = x x`

11. Pasar de notación Haskell a notación de funciones anónimas (llamada *notación lambda*).

- a) la función `smaller` definida por

```
smaller (x,y,z) | x <= y && x <= z = x
              | y <= x && y <= z = y
              | z <= x && z <= y = z
```

b) `second x = \x -> x`

c) la función `andThen` definida por

```
andThen True y = y
andThen False y = False
```

12. Pasar de notación lambda a notación Haskell.

a) `iff = \x -> \y -> if x then not y else y`

b) `alpha = \x -> \x -> x`

### Ejercicios complementarios

13. Definir y dar el tipo de la función que recibe una terna de coeficientes y devuelve las raíces de la ecuación cuadrática correspondiente.

14. Para cada una de las siguientes expresiones decidir si están bien formadas. Si es así dar su valor o en caso contrario especificar si el error es sintáctico, de tipos, o de alguna otra clase:

a) `(3 == --3) && True`

b) `1 && 2 == 2`

c) `1 + if ('a' < 'b') then 3 else 5`

d) `let par = (True,4) in (if first par then first par else second par)`

e) `(a + 23); b + 4`