

# (Dis)funcional

Algunas de las siguientes soluciones no funcionan; otras sí, pero no son funcionales. Detectá el error en cada caso y corregilo:

```
tieneNombreLargo mascota = length (fst mascota) > 9 == True

sumarEnergia (Persona _ energia _ _) = (Persona _ (energia + 5) _ _)

triplicarLosPares numeros = (map (*3) . filter . even) numeros

sonTodosMamiferos animales = all (==True) (map esMamifero animales)

sonTodosMamiferos' animales = (and . map esMamifero) animales

abrirVentanas :: Casa -> Casa
prenderEstufa :: Casa -> Casa
encenderElAireA :: Casa -> Int -> Casa
mudarseA :: String -> Casa -> Casa

miCasaInteligente = Casa
  { direccion = "Medrano 951",
    temperatura = 26,
    reguladoresDeTemperatura = [abrirVentanas, prenderEstufa,
      mudarseA, encenderElAireA 24]
  }

esBeatle _ = False
esBeatle "Ringo" = True
esBeatle "John" = True
esBeatle "George" = True
esBeatle "Paul" = True

sumaDeLasEdadesRecursiva [] = 0
sumaDeLasEdadesRecursiva lista =
  edad (head lista) + sumaDeLasEdadesRecursiva (drop 1 lista)

abrirVentanas casa = casa { direccion = direccion casa, temperatura = temperatura
casa - 2, reguladoresDeTemperatura = reguladoresDeTemperatura casa }

agregarValor valor indice lista =
  take indice lista ++ [valor] ++ drop indice lista

poneleUnNombre numeros = (sum (map (*3) (filter even numeros))) < 100
```