

Cada pregunta tiene (espero) una y solo una respuesta correcta.

1. ¿Cuántas de las siguientes expresiones son sintácticamente equivalentes a $[[1,2],[]]$?

$[1:2:[]]$ $1:2: [[]]$ $[1,2]: [[]]$ $[1:[2],[]]$

- ☒ Exactamente dos ☐ Exactamente tres ☐ Exactamente cuatro

2. En las siguientes expresiones suponemos que los números $1, 2, \dots$ tienen el tipo concreto `Int`. ¿Cuántas de ellas están mal tipadas?

$[1]:[2]$ $[[]]:[]$ $[]:[]$ $(1:2):[]$ $1:(2:[])$

- ☐ Exactamente una de ellas
☒ Exactamente dos de ellas
☐ Las dos anteriores son falsas.

3. Considérense las expresiones de tipo (que solo difieren en los paréntesis):

$\tau_1 = a \rightarrow ((a \rightarrow a) \rightarrow a)$
 $\tau_2 = a \rightarrow (a \rightarrow a) \rightarrow a$
 $\tau_3 = a \rightarrow a \rightarrow a \rightarrow a$

- ☒ $\tau_1 \equiv \tau_2 \neq \tau_3$
☐ $\tau_1 \equiv \tau_3 \neq \tau_2$
☐ $\tau_1 \neq \tau_2 \neq \tau_3 \neq \tau_1$

4. Considérense las expresiones (que solo difieren en los paréntesis):

$e_1 = f \ x \ 1 \ (x + y)$
 $e_2 = (f \ x \ 1) \ (x + y)$
 $e_3 = f \ x \ 1 \ ((+) \ x \ y)$

- ☐ $e_1 \neq e_2 \neq e_3 \neq e_1$
☐ $e_1 \equiv e_3 \neq e_2$
☒ $e_1 \equiv e_2 \equiv e_3$

5. La evaluación de la expresión `foldr (\x y -> y - x) 0 [2,2,2]` produce como resultado:

- ☐ 0 ☒ -6 ☐ 2

6. La evaluación de `foldl (\e x -> x:x:e) [] [1,2,3]` produce como resultado

- ☐ $[1,1,2,2,3,3]$
☒ $[3,3,2,2,1,1]$
☐ $[3,2,1,3,2,1]$

7. En esta pregunta, por simplificar, suponemos que xs, ys son listas finitas. Suponemos, además, que xs, ys, f, g son todas del tipo adecuado para que todas las expresiones estén bien tipadas.

Considérense las afirmaciones: `map id xs = id xs`, para toda xs

`map f (xs ++ ys) = (map f xs) ++ (map f ys)`, para todo f, xs, ys

`map (f.g) xs = (map f . map g) xs`, para todo f, g, xs

- ☐ Exactamente una de las tres afirmaciones es cierta
☐ Exactamente dos de las tres afirmaciones son ciertas
☒ Las tres afirmaciones son ciertas

8. La evaluación de `length [(x,y)|x <- [1,3..9], y <- [1..x], y > x-1]` produce como resultado

- ☐ 0 ☒ 5 ☐ Un error

9. La evaluación de la expresión `(\x y -> y (x y)) (\x -> x 0) (\x -> x+1)` produce como resultado

- ☐ 1 ☒ 2 ☐ Un error

10. Sea f definida por $f \ x \ y = y \ (x \ y)$. El tipo de f es:

- ☐ $a \rightarrow (a \rightarrow b) \rightarrow a \rightarrow b$
☐ $a \rightarrow a \rightarrow (a \rightarrow a)$
☒ $((a \rightarrow b) \rightarrow a) \rightarrow (a \rightarrow b) \rightarrow b$

11. La evaluación de `map (zipWith (*3) [3,2,1]) [[1,2,3],[4,5,6]] !! 1 !! 0` produce como resultado:

- ☐ 9 ☒ Un error de tipos ☐ Un error en tiempo de ejecución

12. Considera las siguientes expresiones:

```
head (take (10^30) ([1..10^30] ++ [1..10^30]))
last (take 1000 ([1..10^30] ++ [1..10^30]))
last ([1..10] ++ [1..10^30])
```

¿Cuántas de ellas nos llevará toda la vida evaluarlas?

- ☒ Exactamente una de ellas
☐ Exactamente dos de ellas
☐ Las tres
-

13. La evaluación de `(head.tail) (map ((take 2).(iterate (+ 3))) [1,3..10])` produce como resultado

- ☐ [4,6,8,11,14]
☐ [1,4]
☒ [3,6]
-

14. Sea f definida por las siguientes ecuaciones: $f \text{ False } y = y$ ¿Cuál de las siguientes afirmaciones es cierta?

$f \text{ x False } = x$
 $f \text{ True True } = \text{False}$

- ☐ La función no es estricta en ninguno de sus dos argumentos
☐ La función es estricta en el primero pero no en el segundo argumento
☒ Las dos anteriores son falsas.
-

15. ¿Cuál de los siguientes valores de e hace que la expresión `takeWhile (\(x,y) -> x>y) (zip (e 1) [0..10])` esté bien tipada?

- ☐ `iterate (+)` ☒ `iterate (+ 3)` ☐ `iterate 3`
-

16. Considérense las expresiones siguientes:

$e_1 \equiv (\text{let } x=5 \text{ in } x+x) + (\text{let } x=3 \text{ in } 2*x)$ $e_2 \equiv \text{let } y=x+x \text{ in let } x=1+x \text{ in } y*y*x$
 $e_3 \equiv \text{let } x=2 \text{ in let } y=x+x \text{ in } y*y*x$ $e_4 \equiv \text{let } \{y=x+x; x=2\} \text{ in } y*y*x$
 $e_5 \equiv [i \mid i <- [1..j], j <- [0..100], \text{mod } j \ 3 == 0]$ $e_6 \equiv [i \mid j <- [0..100], i <- [1..j], \text{mod } j \ 3 == 0]$

La evaluación de esas expresiones dará error por problemas de ámbito de variables:

- ☒ Exactamente en dos de ellas
☐ Exactamente en tres de ellas
☐ Exactamente en cuatro de ellas
-

17. La evaluación de la expresión `let {x = 1:y ; y = 2:x} in x !! 2` produce como resultado:

- ☐ Un error sintáctico o de tipos ☐ Un cómputo no terminante ☒ 1
-

18. Considérense las siguientes definiciones de funciones, donde f, g son dos funciones cualesquiera:

$h \ x \ y = f \ (g \ x) \ y$ $h' = f \ g$ $h'' = f \ . \ g$

- ☐ h, h' y h'' son equivalentes
☒ h y h'' son equivalentes, pero h' no
☐ Las dos anteriores son falsas.
-

19. Sea e una expresión. ¿Cuántas de las siguientes situaciones son posibles al evaluar e ?

- Por evaluación impaciente se obtiene el valor 3, y por evaluación perezosa el valor 2.
 - Se obtiene el valor 3 tanto por evaluación impaciente como por evaluación perezosa.
 - Por evaluación perezosa el cómputo no termina, y por evaluación impaciente el cómputo termina.
 - Por evaluación perezosa el cómputo termina, y por evaluación impaciente el cómputo no termina.
 - El cómputo no termina ni por evaluación impaciente ni por evaluación perezosa.
- ☐ Exactamente dos ☒ Exactamente tres ☐ Las dos anteriores son falsas.
-

20. ¿Cuál de las siguientes definiciones es equivalente a `f n m = [x*y | x <- [1..n], y <- [x..m]]`?

- ☒ `f n m = concat (map f [1..n]) where f x = map (\y -> x*y) [x..m]`
☐ `f n m = concat (map f [x..m]) where f y = map (\y -> x*y) [1..n]`
☐ `f n m = zipWith (*) [1..n] [x..m]`
-