

## Actividad de evaluación de Programación Funcional

|                            |               |
|----------------------------|---------------|
| <b>Nombre y Apellidos:</b> |               |
| <b>UO:</b>                 | <b>Firma:</b> |

1. Evalúa **paso a paso** las siguientes expresiones: **(5 puntos)**

- i. `(map list '(1 2) '(3 4) '(5 6))`
- ii. `((lambda(a . b) (list a b)) 1 2 3)`
- iii. `(filter (curry > 3) '(2 4 6 1))`
- iv. `((lambda a (apply + 5 a)) 1 7)`
- v. `(let [(a 1) (b list?)]  
 (let* [(a 3) (b number?) (c (b a))]  
 (list a b c)))`

2. Responde **justificadamente** a las siguientes cuestiones: **(5 puntos)**

- i. ¿Cuándo una función se considera de orden superior?
- ii. A la vista de las dos siguientes funciones:  
    `maxInt(Lint)::=` retorna el máximo de la lista de enteros `Lint`  
    `minReal(Lreal)::=` retorna el min de la lista de reales `Lreal`

Identifica que patrón tienen en común ambas funciones y enuncia una nueva FOS que cubra las funciones anteriores, y otras similares. A modo de ejemplo, indica como implementarías al menos una de las dos funciones iniciales en base a la nueva función.

- iii. ¿Qué es la evaluación perezosa y que ventaja tiene frente a la evaluación ansiosa?  
**Pon un ejemplo.**
- iv. Con la función **letrec** de Scheme nos permite asociar a un símbolo **x** una expresión que contiene al propio símbolo **x** que estamos definiendo. ¿Qué finalidad tiene? **Pon un ejemplo.**
- v. ¿Qué mecanismo de Programación Funcional permite invocar una función con menos argumentos de los indicados en su definición? **Pon un ejemplo.**