

- i. (map list '(1 2) '(3 4) '(5 6))
((list 1 3 5) (list 2 4 6))
 ((1 3 5) (2 4 6))
- ii. ((lambda(a . b) (list a b)) 1 2 3)
(list 1 '(2 3))
 (1 (2 3))
- iii. (filter (curry > 3) '(2 4 6 1))
(filter (lambda(x) (> 3 x)) '(2 4 6 1))
 (2 1)
- iv. ((lambda a (apply + 5 a)) 1 7)
(apply + 5 (1 7))
(+ 5 1 7)
 13
- v. (let [(a 1) (b list?)]
(let* [(a 3) (b number?) (c (b a))]
(list a b c)))
 ;a=4 b=list?
(let* [(a 3) (b number?) (c (b a))]
(list a b c))
 ;a=5 b=number? c=(number? 5)=>#t
(list 5 number? #t)
 (5 number? #t)
- ```
((lambda(a b)(list a b)) 1 2)
(1 2)
> ((lambda(a b)(list a b)) 1 3 4)
. . arity mismatch;
 the expected number of arguments does not
match the given number
 expected: 2
 given: 3
> ((lambda(a . b)(list a b)) 1 3 4)
(1 (3 4))
```
- ^^ diferencias ^^