

Guía de Trabajos Prácticos 2 – Programación Funcional

1. Reescriba las siguientes expresiones, usando Let para remover las subexpresiones comunes y para mejorar la estructura del código. No realice ninguna simplificación algebraica.

a. $(+ (/ (* 7 a) b) (/ (* 3 a) b) (/ (* 7 a) b))$

b. $(\text{cons} (\text{car} (\text{list } a \ b \ c)) (\text{cdr} (\text{list } a \ b \ c)))$

2. Determine el valor de la siguiente expresión. Explique como ha derivado este valor.

```
(let ((x 9))
  (* x
    (let ((x (/ x 3)))
      (+ x x))))
```

3. Reescriba las siguientes expresiones para darles un único nombre a cada variable let-bound diferente de forma que ninguna quede *ensombrecida*. Verifique que el valor de su expresión es el mismo que el de la expresión original.

a. $(\text{let} ((x 'a) (y 'b))$
 $(\text{list} (\text{let} ((x 'c)) (\text{cons } x \ y))$
 $(\text{let} ((y 'd)) (\text{cons } x \ y))))$

b. $(\text{let} ((x '(\text{a } b \ c)))$
 $(\text{cons} (\text{let} ((x (\text{cdr } x)))$
 $(\text{car } x))$
 $(\text{let} ((x (\text{car } x)))$
 $(\text{cons} (\text{let} ((x (\text{cdr } x)))$
 $(\text{car } x))$
 $(\text{cons} (\text{let} ((x (\text{car } x)))$
 $x)$
 $(\text{cdr } x))))))$

4. Determine el valor de las siguientes expresiones.

a. $(\text{let} ((f (\text{lambda } (x) x)))$
 $(f 'a))$

b. $(\text{let} ((f (\text{lambda } x \ x)))$
 $(f 'a))$

c. $(\text{let} ((f (\text{lambda } (x \ . \ y) x)))$
 $(f 'a))$

d. $(\text{let} ((f (\text{lambda } (x \ . \ y) y)))$
 $(f 'a))$

5. Realice una función de una operación aritmética simple (+, -) y la convierta en sintaxis de Scheme.

6. El procedimiento **length** retorna la longitud de su argumento, que debe ser una lista. Por ejemplo, (**length** '(a b c)) es 3. Usando **length**, defina el procedimiento **shorter**, que retorna la lista más corta de los dos argumentos pasados o la primera lista si tienen el mismo largo.

(shorter '(a b) '(c d e)) \Rightarrow (a b)

(shorter '(a b) '(c d)) \Rightarrow (a b)

(shorter '(a b) '(c)) \Rightarrow (c)

7. Defina una función que devuelva el área de un círculo.

Ej: (area 3) \rightarrow 28.26

8. Defina una función que devuelva el perímetro de un círculo.

Ej: (perímetro 1) \rightarrow 6.28

9. Defina una función llamada **circlestuff** que compute el área y el perímetro de un círculo y los devuelva en una lista, con el área primero.

Ej: (circlestuff 3) \rightarrow (28.26 18.84)

10. Defina una función **distance2d** que reciba como parámetros dos puntos en el plano y devuelva su distancia. Utilice una lista impropia para la declaración de x e y.

Ej: (distance2d '(1 . 1) '(2 . 2)) \rightarrow 1.41

11. Defina una función **próximo** que reciba un punto en el plano y una lista de puntos y devuelva la distancia al más cercano.