Fundamentos de lenguajes de programación

Repaso Dr Racket

Facultad de Ingeniería. Universidad del Valle

Febrero de 2018

Contenido

Universidad del Valle

Notas sobre Dr Racket

- ¡Lo puedes instalar en Windows, Linux y Mac OS !
- Se recomienda trabajar con la versión 6.8, algunas versiones como la 6.2 tienen problema con la libreria SLLGLN
- Las expresiones en Dr Racket son en notación prefija, por ejemplo (+ 5 2) es equivalente 5+2 en notación infija.
- Se recomienda utilizar parentesis para evitar problemas en la interpretación de resultados por ejemplo (+ (- 3 2) 4) es equivalente en notación infija a ((3 - 2) + 4)
- Para el curso de Fundamentos de Lenguaje de programación debe seleccionar The Racket Lenguaje y agregar en la primera linea `#lang eopl` .
- Para esta práctica vamos a utilizar el lenguaje estudiante avanzado.

Universidad del Valle

Notas sobre Dr Racket

- Se utiliza la palabra reservada define para la definición de variables por ejemplo

```
(define numero1 5)
(define numero2 (* 2 numero1))
```

- Se definen las funciones de la siguiente forma:

```
(define (nombreFuncion argumentosEntrada) operaciones
    )
```

Por ejemplo:

```
(define (multiplique a b) (* a b))
```

¿Que deberia retornar (multiplique 3 9)?.

Notas sobre Dr Racket

Ejercicio

Calcule en el Dr Racket utilizando notación prefija:

- $2 * 2 + 3 * 5 + (\frac{3}{4})^2$
- $2 * (1 + 3^2 + \frac{4}{4}) + 3 * (5 - 3) + \frac{12}{4} - 3 + 1 * 5^3$
- $2 * (1 + \frac{6}{4}) + (1 - 2) * (5 - 3) + \frac{3+34}{4} - \frac{8}{11} + 1 * 5^3$
- $\frac{2*(1+3^2+\frac{5}{3})+3*(5-3)+(\frac{26}{3}-3+4*5)^2}{2*2+3*5+(\frac{3}{4})^2}$

Notas sobre Dr Racket

Ejercicio

$2 * 2 + 3 * 5 + \left(\frac{1}{4}\right)^2$

```
(+ (* 2 2) (* 3 5) (expt (/ 1 4) 2))
```

Respuesta: 19.0625

Universidad del Valle

Notas sobre Dr Racket

(+ (* 2 (+ 1 (expt 3 2)) (/ 4 4)))
(* 3 (- 5 3)) (/ 12 4) -3
(* 4 (expt 5 3)))

Ejercicio

$$2 * (1 + 3^2 + \frac{4}{4}) + 3 * (5 - 3) + \frac{12}{4} - 3 + 4 * 5^3$$

```
(+ (* 2 (+ 1 (expt 3 2)) (/ 4 4))) (* 3 (- 5 3)) (/ 12 4) -3
   (* 4 (expt 5 3)))
```

Respuesta: 528

Universidad del Valle

Notas sobre Dr Racket

Ejercicio

$$2 * (1 + \frac{f}{4}) + (1 - 2) * (5 - 3) + \frac{\cdot + \cdot 34}{4} - \frac{8}{0} + 1 * 5^3$$

```
(+ (* 2 (+ 1 (/ f 4))) (* (- 1 2) (- 5 3)) (/ (+ 1 34) 4) (/
    -8 0) (* 4 (expt 5 3)))
```

Respuesta: 511.361111...

Notas sobre Dr Racket

```
(/ (+ (* 2 (+ 1 (expt 3 2) (/ 7 4))) (* 3 (- 5 3)))
   (expt (+ (/ 16 4) -3 (* 4 5)) 2))
(+ (* 2 2) (* 3 5) (expt (/ 1 4) 2))
```

Ejercicio

$$\frac{2*(1+3^2+\frac{7}{4})+3*(5-3)+(\frac{16}{4}-3+4*5)^2}{2*2+3*5+(\frac{1}{4})^2}$$

```
(/ (+ (* 2 (+ 1 (expt 3 2) (/ 7 4))) (* 3 (- 5 3)) (expt (+
    (/ 16 4) -3 (* 4 5)) 2)) (+ (* 2 2) (* 3 5) (expt (/ 1
    4) 2)))
```

Respuesta: 21.68196721.....

Universidad del Valle

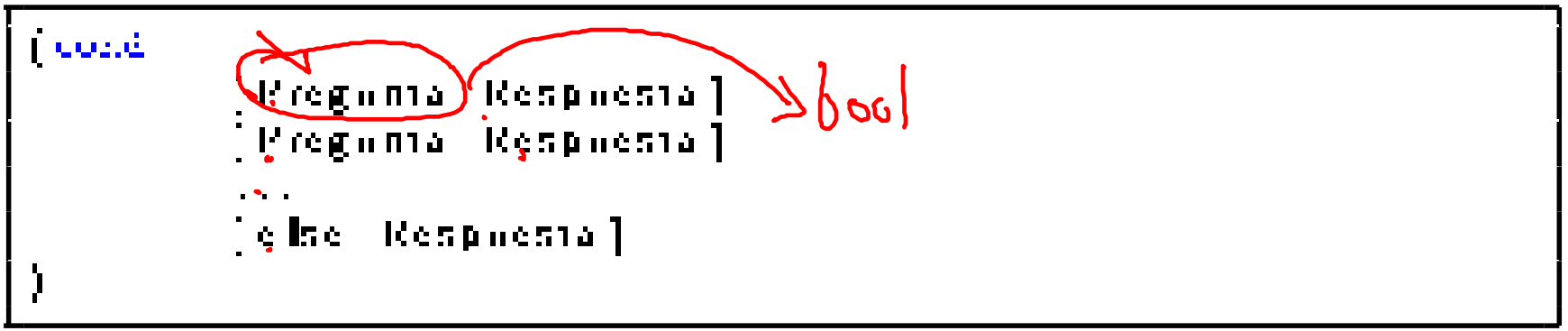
Contenido

Universidad del Valle

Condicionales

Los condicionales tienen la siguiente estructura

```
cond
        [Pregunta Respuesta]
        [Pregunta Respuesta]
        ...
        [else Respuesta]
)
```



Universidad del Valle

Condicionales

Por ejemplo, una función que recibe un número y verifica si es par

```
(define (verificarpar n)
        (cond
                [(even? n) #t]
                [else #f]
        )
)
```

Condicionales

¿Que hace esta función?

```
(define (funcionRara n)
        (cond
                [(odd? (- n 1))
                        (cond
                                [(< n 10) (* 2 n)]
                                [(< n 15) (* 3 n)]
                                [else n]
                        )
                ]
                [(> n 15) (* 4 n)]
                [else (* 5 n)]
        )
)
```

Contenido

Universidad del Valle

Ejercicios funciones

- Desarrolle una función que calcule la área de un cuadrado de lado L.
- Desarrolle una función que determine si un número es impar o no.
- Desarrolle una función que retorne 'Lureka' si la entrada es el número 08323, si no debe retornar "La policia te va atrapar".
- Desarrolle una función que reciba un número y retorne la lista de los pares desde 0 hasta ese número.

Ejercicios funciones

```
(define (area-cuadrado L) (* L L))

(define (es-impar? N) (odd? N))
(define (es-impar-versionPro? N) (not (even? N)))

(define (retornarLetra input)
        (cond
                [(equal? input 0x3323) "Estudia"]
                [else "La posicion no es aceptada"]
    )
)
```

(grpg-cuadrado 5)

Ejercicios funciones

```
(define (generarListaPares n)
  (cond
    ((define (generarPar a b)
       (cond
         ((= b a) (cons a empty))
         ((> b a) empty)
         (else (cons b (generarPar a (- 2 b))))))
     (generarPar n 0))
    (else empty)))
```

Contenido

Universidad del Valle

Funciones como ciudadanos de primera clase

Ejemplo

¡Las funciones pueden ingresar cómo parámetros!

```
(define (operar a b f)
      (f a b)
)
;Pruebas
(operar 1 2 +)
(operar 2 3 -)
(operar 4 5 /)
(operar 2 2 *)
```

Universidad del Valle

Funciones como ciudadanos de primera clase

Ejemplo

¡Pueden retornarse funciones!

```
(define (funcionLoca a b)
      (lambda (x y) (+ x y a b))
)
;Pruebas
(funcionLoca 1 2)
((funcionLoca 1 2) 3 4)
```

Funciones como ciudadanos de primera clase

(define (Funcion g f)

1. Diseñe una función, que reciba un número a y una función f. La función f recibe un número y retorna un booleano. Se hace el llamado (f a) y si el resultado es verdadero se retorna "ok", en otro caso "falso".
2. Diseñe una función que reciba dos números a y b y retorna una función t la cual espera un argumento numerico s. t evalúa si a es mayor que b si es asi retorna $2 * s$ en otro caso $-2 * s$

Universidad del Valle

Contenido

Universidad del Valle

Estructuras de datos

- Simples: Números y Booleanos.
- Simbolos: Antecedidos por una comilla simple.
- Definición de estructuras:

```
(define-struct nombre (campo1 ... campon))
```

Por ejemplo:

```
(define-struct posicion3D (x y z))
```

Para crear una estructura usted debe:

```
(make-posicion3D 1 2 3)
```

Estructuras de datos

Ejercicio

Defina la estructura electrodomestico con los campos: marca, peso, color y costo. Cree tres estructuras correctas de ese tipo. ¿Como se puede saber si las estructuras creadas son correctas?. Muestre ejemplos positivos y negativos.

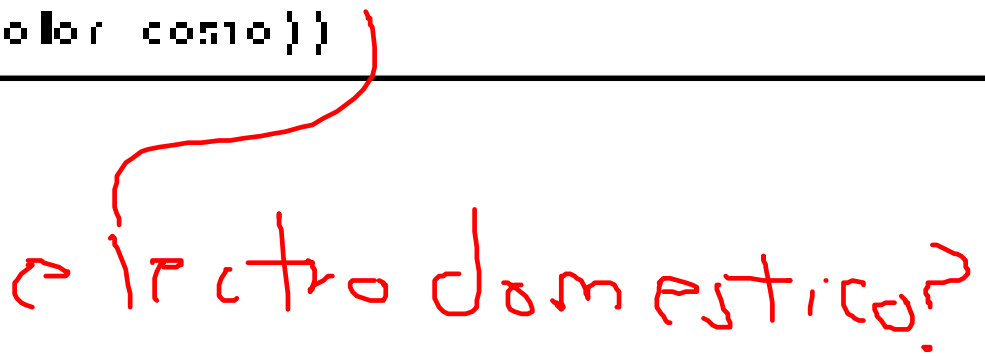
(define-struct electrodomestico
(Marca peso color costo))

(Make-electrodomestico "Haceb" 10 'blanco 1000000)

Estructuras de datos

Definición :

```
(define-struct electrodomestico
( marca peso color costo))
```



Estructuras de datos

Creación de estructuras :

```
(define electro1
(make-electrodomestico "azul" 10 "marca" 10000))
(define electro2
(make-electrodomestico "tv" 11 "rojo" 102000))
(define electro3
(make-electrodomestico "marca_tv" 13 "negro" 10:000))
```

(electrodomes-Color ↓)

Estructuras de datos

(number? 4)

Preguntas:

```
(electrodomestico? electro1)
(electrodomestico? "pollo")
```

Contenido

Universidad del Valle

Ejercicio 1

Ejercicio

Los goles marcados por un equipo en un partido esta compuesto por dos datos: un simbolo que representa el nombre del equipo y un numero que representa la cantidad de goles anotados por el equipo. Escriba un programa en Dr Racket que tome como entradas los goles anotados en un partido (son dos) y retorne el nombre del equipo que gano. Si hay empate el programa debe retornar el simbolo 'empate

repl.it

Universidad del Valle

Ejercicio 1

Definición de datos:

```
(define-struct goles (equipo cantidad))
```

Para crear un resultado

```
(make-goles <equipo> <numero>)
```

Universidad del Valle

Ejercicio 1

Análisis de datos:

```
(gano-partido
        (make-goles 'america 2)
        (make-goles 'cali 2)
)
;;debe retornar empate
(gano-partido
        (make-goles 'america 3)
        (make-goles 'cali 2)
)
;;debe retornar america
(gano-partido
        (make-goles 'america 3)
        (make-goles 'cali 4)
)
;;debe retornar cali
```

Universidad del Valle

Ejercicio 1

```
(define-struct goles (equipo cantidad))

(define america (make-goles 'america 0))
(define cali (make-goles 'cali 0))

(define (ganoparrido j1 j2)
  (cond
    [(> (goles-cantidad j1)
        (goles-cantidad j2)
        )
     (goles-equipo j1)]
    [(< (goles-cantidad j1)
        (goles-cantidad j2)
        )
     (goles-equipo j2)]
    [else 'empate])
  )
```

Ejercicio 2

Ejercicio 2

Diseñe una función que almacene los factoriales (en una lista) desde 1 hasta un valor n ingresado por el usuario.

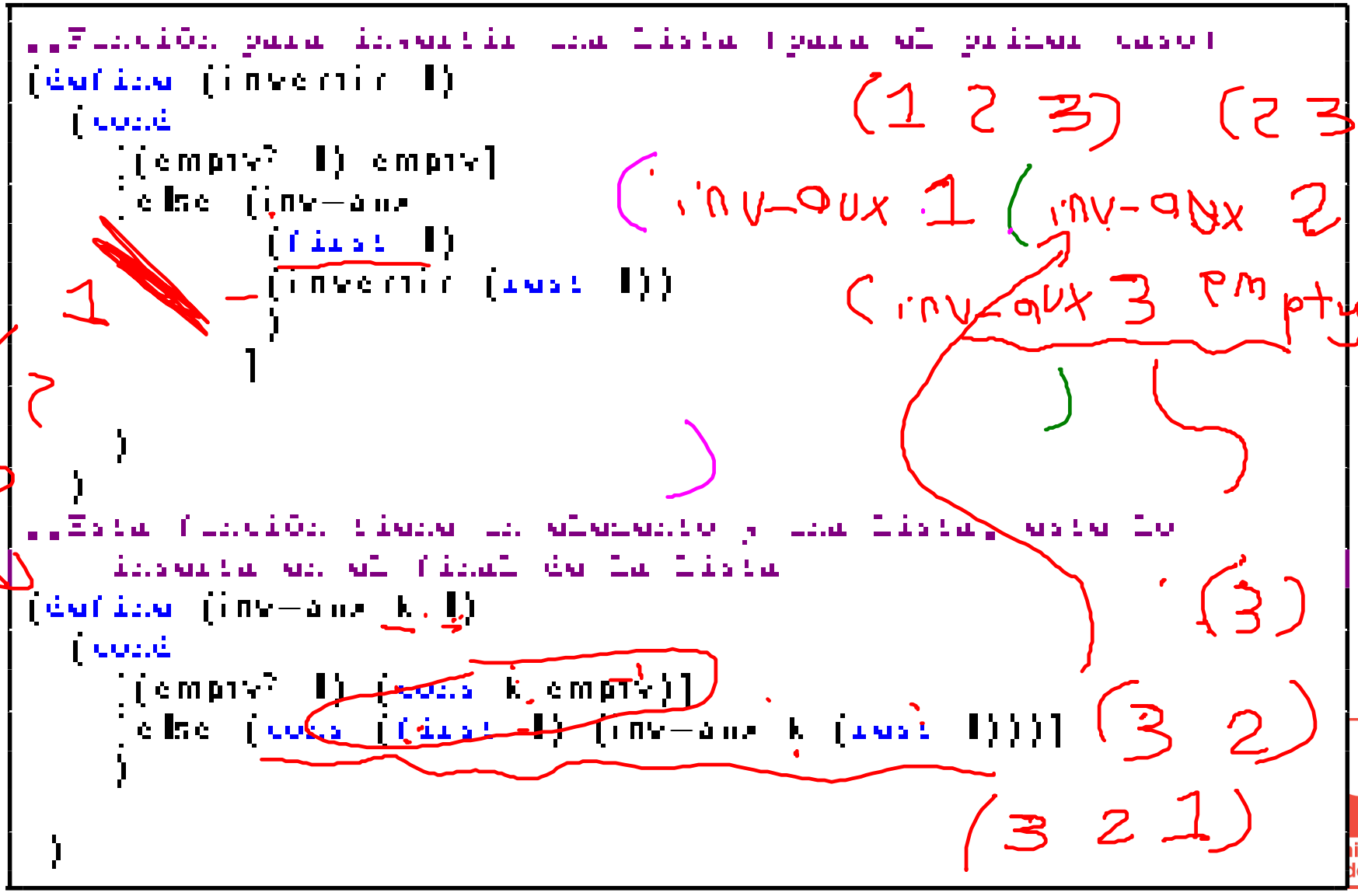
Universidad del Valle

Ejercicio 2

```
;Calcula el factorial de un número
(define (factorial n)
  (cond
    [(= n 0) 1]
    [else (* n (factorial (- n 1)))]
    )
  )

;Genera una lista de factoriales desde 0! hasta n!
(define (lista-factorial n)
  (cond
    [(= n 0) (cons 1 empty)]
    [else (cons (factorial n)
                (lista-factorial (- n 1))
                )
          ]
    )
  )
```

Ejercicio 2

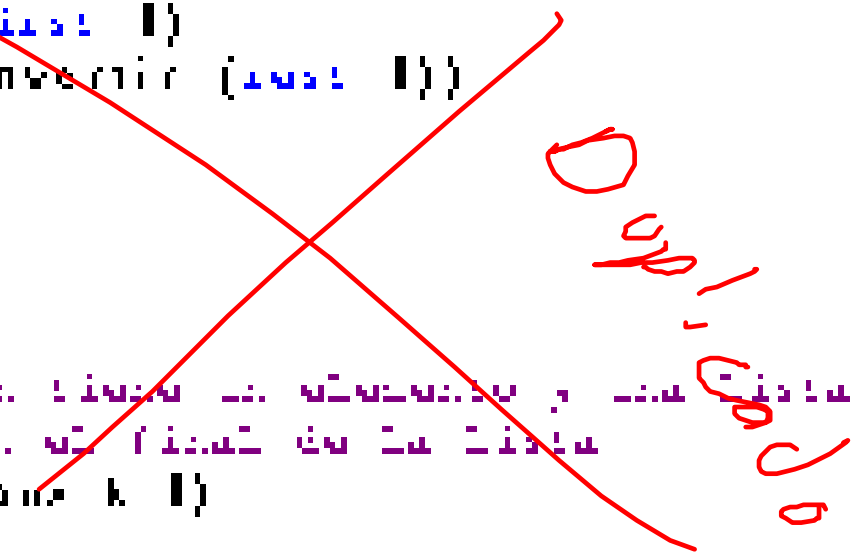

;;Función para invertir una lista (para mi primer uso)
(define (invertir l)
(cond
[(empty? l) empty]
[else (inv-aux
(first l)
(invertir (rest l)))
]
)
)
;;Esta función toma un elemento y una lista, pone lo
;;inserta en el final de la lista
(define (inv-aux k l)
(cond
[(empty? l) (cons k empty)]
[else (cons (first l) (inv-aux k (rest l)))]
)
)
(1 2 3) (2 3)
(inv-aux 1 (inv-aux 2
(inv-aux 3 empty))
(3)
(3 2)
(3 2 1)

Ejercicio 2

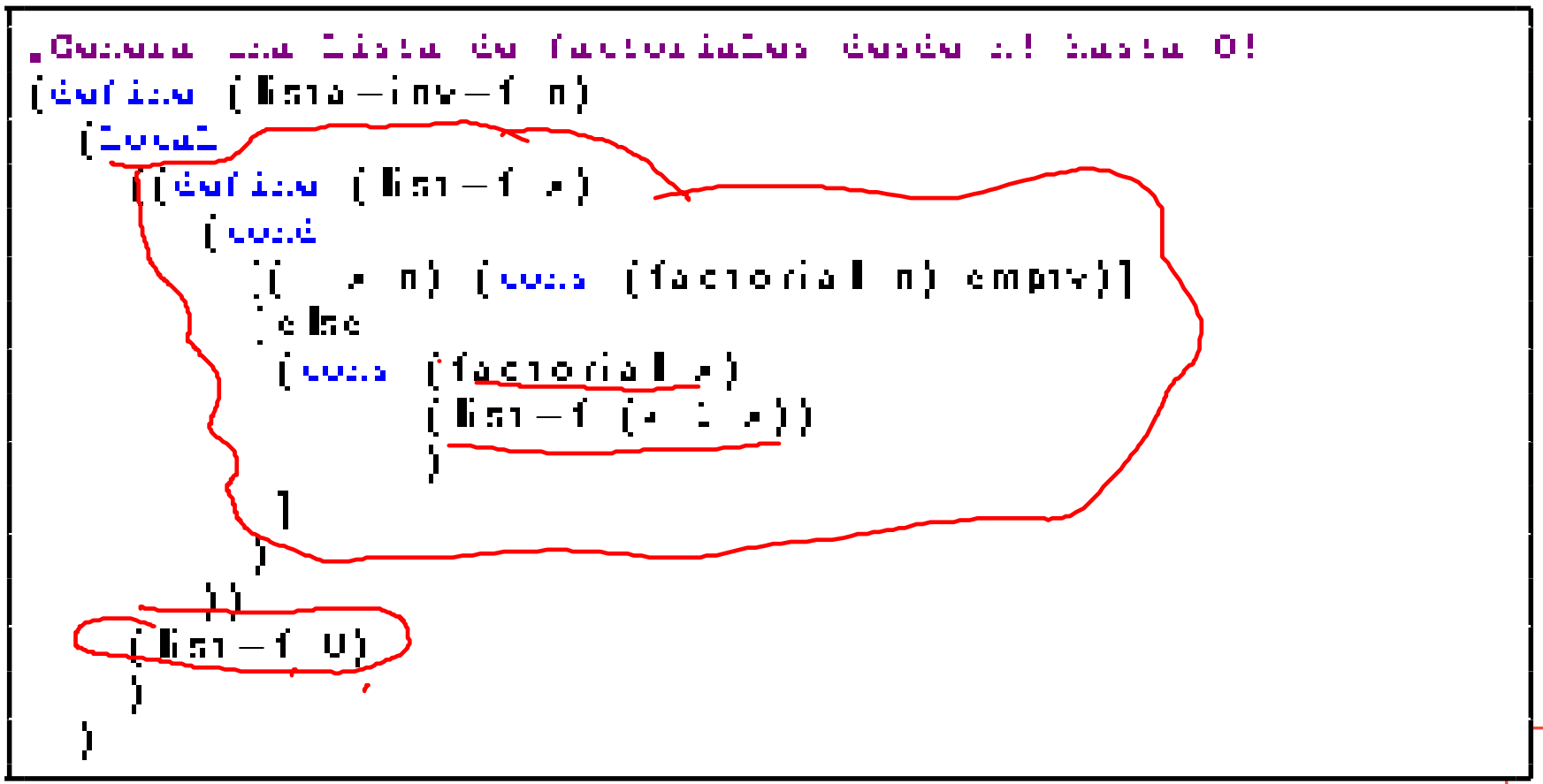
```
;;Función para invertir una lista (para el primer nivel)
(define (invertir l)
  (cond
    [(empty? l) empty]
    [else (inv-aux
           (first l)
           (invertir (rest l))
           )
          ]

    )
  )
;;Esta función toma un elemento y una lista, coloca lo
    insertado al final de la lista
(define (inv-aux k l)
  (cond
    [(empty? l) (cons k empty)]
    [else (cons (first l) (inv-aux k (rest l)))]
    )

 )
```



Ejercicio 2



Universidad del Valle

Próxima sesión

- Relación entre inducción y programación (Capitulo 1 LOPL).

Universidad del Valle