

# Fundamentos de lenguajes de programación

Repaso Dr Racket

[carlos.andres.delgado@correounivalle.edu.co](mailto:carlos.andres.delgado@correounivalle.edu.co)

Carlos Andrés Delgado S.

Facultad de Ingeniería. Universidad del Valle

Febrero de 2017

## Fundamentos de lenguajes de programación

Carlos  
Andrés  
Delgado S.

## Preliminares

## Condicionales

## Funciones

## Funciones como ciudadanos de primera clase

## Estructuras de datos

## Ejercicios

## 1 Preliminares

## 2 Condicionales

## 3 Funciones

## 4 Funciones como ciudadanos de primera clase

## 5 Estructuras de datos

## 6 Ejercicios

# Notas sobre Dr Racket

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

- ¡Lo puedes instalar en Windows, Linux y Mac OS !
- Se recomienda trabajar con la versión 6.7, algunas versiones como la 6.2 tienen problema con la librería SLLGEN
- Las expresiones en Dr Racket son en notación prefija, por ejemplo  $(+ 5 2)$  es equivalente  $5+2$  en notación infija.
- Se recomienda utilizar paréntesis 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 Language** y agregar en la primera línea `#lang eopl` .
- Para esta práctica vamos a utilizar el lenguaje estudiante avanzado.

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

```
(define numeroA 5)  
(define numeroB (* 2 numeroA))
```

- Se definen las funciones de la siguiente forma:

```
(define (nombreFuncion argumentosEntrada) <  
  operaciones >)
```

Por ejemplo:

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

¿Que debería retornar (multiplique 3 9)?.

## Ejercicio

Calcule en el Dr Racket utilizando notación prefija:

- $2 * 2 + 3 * 5 + (\frac{1}{4})^2$
- $2 * (1 + 3^2 + \frac{4}{4}) + 3 * (5 - 3) + \frac{12}{4} - 3 + 4 * 5^3$
- $2 * (1 + \frac{7}{4}) + (1 - 2) * (5 - 3) + \frac{1+34}{4} - \frac{8}{9} + 4 * 5^3$
- $$\frac{2*(1+3^2+\frac{7}{4})+3*(5-3)+(\frac{16}{4}-3+4*5)^2}{2*2+3*5+(\frac{1}{4})^2}$$

## Ejercicio

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

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

**Respuesta:** 19.0625

## 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**

## Ejercicio

$$2 * \left(1 + \frac{7}{4}\right) + (1 - 2) * (5 - 3) + \frac{1+34}{4} - \frac{8}{9} + 4 * 5^3$$

```
(+ (* 2 (+ 1(/ 7 4))) (* (- 1 2) (- 5 3)) (/ (+ 1 34) 4)
  (/ -8 9) (* 4 (expt 5 3)))
```

**Respuesta:** 511.361111...



## 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:** 24.68196721.....

## Fundamentos de lenguajes de programación

Carlos  
Andrés  
Delgado S.

### Preliminares

### Condicionales

### Funciones

### Funciones como ciudadanos de primera clase

### Estructuras de datos

### Ejercicios

## 1 Preliminares

## 2 Condicionales

## 3 Funciones

## 4 Funciones como ciudadanos de primera clase

## 5 Estructuras de datos

## 6 Ejercicios

Los condicionales tienen la siguiente estructura

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

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

```
(define (verificarpar n)
  (cond
    [(even? n) #T]
    [else #F]
  )
)
```

¿Que hace esta función?

```
(define (funcionRara n)
  (cond
    [(odd? (- n 1))
     (cond
       [(< n 10) (* 2 n)]
       [(< n 15) (* 3 n)]
       [else n]
      )
    ]
    [(> n 115) (* 4 n)]
    [else (* 5 n)]
  )
)
```

## Fundamentos de lenguajes de programación

Carlos  
Andrés  
Delgado S.

### Preliminares

### Condicionales

### Funciones

### Funciones como ciudadanos de primera clase

### Estructuras de datos

### Ejercicios

1 Preliminares

2 Condicionales

3 Funciones

4 Funciones como ciudadanos de primera clase

5 Estructuras de datos

6 Ejercicios

- 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 'Eureka' si la entrada es el número 08323, si no debe retornar "La policía te va atrapar".
- Desarrolle una función que reciba un número y retorne la lista de los pares desde 0 hasta ese número.

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

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

(define (retornarTexto input)
  (cond
    [(equal? input 08323) "Eureka" ]
    [ else "La policía te va atrapar" ]
  )
)
```



# Ejercicios funciones

## Fundamentos de lenguajes de programación

Carlos  
Andrés  
Delgado S.

## Preliminares

## Condicionales

## Funciones

## Funciones como ciudadanos de primera clase

## Estructuras de datos

## Ejercicios

```
(define (generarListaPares x)
  (local
    ((define (generarPar a b)
      (cond
        [(= b a) (cons a empty)]
        [(> b a) empty]
        [else (cons b (generarPar a (+ 2 b)))]
      )
    )
    (generarPar x 0)
  )
)
```

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

1 Preliminares

2 Condicionales

3 Funciones

4 Funciones como ciudadanos de primera clase

5 Estructuras de datos

6 Ejercicios

# Funciones como ciudadanos de primera clase

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

## Ejemplo

¡Las funciones pueden ingresar cómo parámetros!

```
(define (operar a b f)
  (f a b))

; Probar
(operar 1 2 +)
(operar 2 3 -)
(operar 4 5 /)
(operar 2 2 *)
```

# Funciones como ciudadanos de primera clase

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

## Ejemplo

¡Pueden retornarse funciones!

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

- 1 Diseñe una función, que reciba un número  $a$  y una función  $f$ . La función  $f$  recibe un número y retorna un booleano. Se debe aplicar  $f$  a  $n$  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 numérico  $s$ .  $t$  evalúa si  $a$  es mayor que  $b$  si es así retorna  $2 * s$  en otro caso  $-2 * s$

## Fundamentos de lenguajes de programación

Carlos  
Andrés  
Delgado S.

### Preliminares

### Condicionales

### Funciones

### Funciones como ciudadanos de primera clase

### Estructuras de datos

### Ejercicios

## 1 Preliminares

## 2 Condicionales

## 3 Funciones

## 4 Funciones como ciudadanos de primera clase

## 5 Estructuras de datos

## 6 Ejercicios

- Simples: Números y Booleanos.
- Símbolos: 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)
```

## Ejercicio

Defina la estructura electrodoméstico 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.



## Definición :

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

## Creación de estructuras :

```
(define electro1  
  (make-electrodomestico "sony" 10 "verde" 10000))  
(define electro2  
  (make-electrodomestico "lg" 11 "rojo" 102000))  
(define electro3  
  (make-electrodomestico "samsung" 13 "negro" 101000))
```

## Preguntas:

```
(electrodomestico? electro1)  
(electrodomestico? "perro")
```

## Fundamentos de lenguajes de programación

Carlos  
Andrés  
Delgado S.

### Preliminares

### Condicionales

### Funciones

### Funciones como ciudadanos de primera clase

### Estructuras de datos

### Ejercicios

## 1 Preliminares

## 2 Condicionales

## 3 Funciones

## 4 Funciones como ciudadanos de primera clase

## 5 Estructuras de datos

## 6 Ejercicios

# Ejercicio 1

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

## Ejercicio

Los goles marcado por un equipo en un partido esta compuesto por dos datos: un símbolo 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 símbolo 'empate

# Ejercicio 1

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

Definición de datos:

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

Para crear un resultado

```
(make-goals <equipo> <puntos>)
```

# Ejercicio 1

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

## 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
```

# Ejercicio 1

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

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

(define america (make-goles 'america 0))
(define cali (make-goles 'cali 10))

(define (ganopartido 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

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

## Ejercicio 2

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

## Ejercicio 2

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

```
;Calcula el factorial de un numero
(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

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

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

;;Esta función tiene un elemento y una lista, este 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)))]
  )
)
```

## Ejercicio 2

Fundamentos  
de lenguajes  
de programación

Carlos  
Andrés  
Delgado S.

Preliminares

Condicionales

Funciones

Funciones  
como  
ciudadanos  
de primera  
clase

Estructuras  
de datos

Ejercicios

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

;;Esta función tiene un elemento y una lista, este 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)))]
  )
)
```

## Ejercicio 2

### Fundamentos de lenguajes de programación

Carlos  
Andrés  
Delgado S.

### Preliminares

### Condicionales

### Funciones

### Funciones como ciudadanos de primera clase

### Estructuras de datos

### Ejercicios

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

- Relación entre inducción y programación (Capítulo 1 EOPL).