



# Taller 1 Racket y recursividad Parte 2

## Fundamentos de Lenguajes Programación

Carlos Andres Delgado S  
carlos.andres.delgado@correounivalle.edu.co

Marzo de 2023

### 1. Reglas

- Se permiten grupos de 2 a 4 personas
- Debe usarse recursión para resolver los problemas
- Debe usarse sintaxis de Racket paréntesis en caso de entregar en papel, en computador debe ser funcional en lenguaje Racket.
- Las funciones deben definirse con **lambda**

#### 1.1. Indicadores de logro

1. IL 1.1.1 Construye datos recursivamente utilizando métodos de inducción y gramáticas BNF
2. IL 1.1.2 Utiliza gramáticas BNF para guiar la construcción de programas recursivos

### 2. Enunciado del taller

De acuerdo a la gramática de listas:

```
<arbol-b> ::= <numero>  
          ::= <simbolo> <arbol-b><arbol-b>
```

1. Diseñar la función **super-mapeo** esta función recibe una lista de funciones que reciben un número y retornan el resultado de un operación (salida numero) y un árbol. Esta función retorna una lista de árboles, cada uno como el resultado de aplicar la operación correspondiente a sus hojas (números), la definición de la función a construir es:

```
(define super-mapeo  
  (lambda ....
```

Ejemplo:

```

(define arbol1
  '(
    x
    (y (z 1 2) (p 2 3))
    (w (a (a 9 10) 13) (t 1 2))
  )
)

(define lf
  (list (lambda (x) (* x 2))
        (lambda (x) (+ x 2))
        (lambda (x) (/ x 8))
  )
)

( super-mapeo lf arbol1 )

```

La salida debe ser:

```

(
  (x (y (z 2 4) (p 4 6)) (w (a (a 18 20) 26) (t 2 4)))
  (x (y (z 3 4) (p 4 5)) (w (a (a 11 12) 15) (t 3 4)))
  (x (y (z 1/8 1/4) (p 1/4 3/8)) (w (a (a 1 1/8 1 1/4) 1 5/8) (t 1/8 1/4)))
)

```

2. Diseñar la función **busqueda-increible-arbolea** esta función recibe dos árboles y una función que recibe dos números y retorna un número, esta retorna una lista de los valores calculados en las hojas que conservan la misma estructura.

```

(define busqueda-increible-arbolea
  (lambda ...

```

Definimos un segundo árbol como ejemplo.

```

(define arbol2
  '(
    a
    (b (d 1 2) 2)
    (c (a 2 13) (c 1 4))
  )
)

;;Llamamos la función
( busqueda-increible-arbolea
  arbol1
  arbol2
  (lambda (x y) (+ x y 2)))

```

Retorna

```

(4 6 28 4 8)

```