



**Implementación de métodos computacionales
(TC2037.2)**

Actividad 2.2

Programación funcional, parte 2

Profesor:

Alejandro de Gante

Alumnos:

Aldo Alejandro Degollado Padilla	A01638391
Abraham Mendoza Pérez	A01274857
Luis Alonso Martínez García	A01636255

Jueves 04 de marzo de 2021

;1. La funcion insert toma dos entradas: un numero n y una lista lst que contiene numeros en orden ascendente.

;Devuelve una nueva lista con los mismos elementos de lst pero con n insertado en su lugar correspondiente.

```
(define (insert n lst)
  (cond
    ((null? lst) (list n))
    ((>= (car lst) n) (cons n lst))
    (else
     (cons (car lst) (insert n (cdr lst))))))
```

; 3. La funcion deep-reverse toma una lista como entrada. Devuelve una lista con los mismos elementos que su

; entrada pero en orden inverso. Si hay listas anidadas, estas tambien se invierten.

```
(define (rotate-left n lst)
  (cond ((null? lst) lst)
        ((= n 0) lst)
        ((negative? n) (rotate-left (+ n 1) (append (list(car (reverse lst))) (reverse (cdr (reverse lst))))))
        (else (rotate-left (- n 1) (append (cdr lst) (list (car lst)))))))
```

; 6. La función rotate-left toma dos entradas: un número entero n y una lista lst. Devuelve la lista que resulta de rotar lst un total

; de n elementos a la izquierda. Si n es negativo, rota hacia la derecha.

```
(define (deep-reverse lst)
  (define (deep-reverseR lst acc)
    (if (null? lst)
```

```
acc
(if (list? (car lst))
    (deep-reverseR (cdr lst) (cons (deep-reverse (car lst)) acc))
    (deep-reverseR (cdr lst) (cons (car lst) acc))
)
)
)
(deep-reverseR lst '())
)
```