

RECURSIVIDAD EN FUNCIONES

1. Invertir los elementos de una lista

```
CL-USER 1 > (defun invertir (lista) (if lista (append (last lista) (invertir (butlast lista)))))
```

INVERTIR

```
CL-USER 2 > (invertir '(a b c))
```

(C B A)

2. DEVUELVE LOS NUMEROS PARES DE UNA LISTA

```
CL-USER 3 > (defun par(lista)
```

```
  (cond ( (null lista) nil )
```

```
        ( (evenp(car lista)) (cons(car lista) (par(cdr lista))) )
```

```
        ( t(par(cdr lista)))
```

```
  )
```

```
)
```

PAR

```
CL-USER 4 > (par '(2 43 50))
```

3. Función que devuelve los números cuadrados de una lista

```
CL-USER 1 > (defun cuadrado (c) (if (null c) nil (cons (expt (car c) 2) (cuadrado (cdr c)))))
```

CUADRADO

```
CL-USER 2 > (cuadrado '( 5 4 7 8))
```

(25 16 49 64)

4. Función que cuenta los elementos de una lista

```
CL-USER 1 > (defun conteo(lista) (cond ((null lista)0)((atom lista)1)( t (+ (conteo(car lista))(conteo(cdr lista)) ) ) )
```

```
CL-USER 9 > (conteo '(3 6 8 9))
```

4

5. BUSQUEDA BINARIA (BUSCA SI UN ELEMENTO PERTENECE A UNA LISTA)

```
(defun busbin(lista n)
  (if (endp lista) nil
      (if (= n (car lista)) lista
          (busbin (cdr lista) n)
      )))
```