

# Examen de Práctica Lisp – Listas y Condicionales

Eduardo Alcaraz

September 24, 2025

## Instrucciones

Responde a cada ejercicio escribiendo las funciones en Lisp.

1. Explica la diferencia entre usar ‘if’, ‘cond’, ‘when’ y ‘unless’.
2. ¿Qué devuelven ‘car’ y ‘cdr’? ¿Cómo se pueden combinar para obtener elementos intermedios de una lista?

## Ejercicio 1 – N-ésimo elemento con car/cdr

Escribe una función ‘(n-esimo n lista)’ que devuelva el **n-ésimo elemento** de una lista utilizando solo ‘car’ y ‘cdr’.

Ejemplo:

```
(n-esimo 3 '(a b c d e)) ;; => c
```

## Ejercicio 2 – Filtrar positivos con when

Escribe una función ‘(filtra-positivos lista)’ que reciba una lista de números y devuelva una nueva lista con solo los números positivos. Usa ‘when’ dentro de un ‘mapcar’ o ‘loop’.

Ejemplo:

```
(filtra-positivos '(-2 0 3 -5 7)) ;; => (3 7)
```

## Ejercicio 3 – Clasificación con cond

Escribe una función ‘(clasifica-numero n)’ que:

- Devuelva “Negativo” si  $n < 0$
- Devuelva “Cero” si  $n = 0$
- Devuelva “Pequeño” si  $1 \leq n \leq 10$
- Devuelva “Mediano” si  $11 \leq n \leq 100$
- Devuelva “Grande” si  $n > 100$

Ejemplo:

```
(clasifica-numero 57) ;; => "Mediano"
```

## Ejercicio 4 – Suma de pares con unless

Escribe una función '(suma-pares lista)' que:

- Devuelva la suma de todos los números pares en la lista.
- Ignore los impares usando 'unless'.

Ejemplo:

```
(suma-pares '(1 2 3 4 5 6)) ;; => 12
```

## Ejercicio 5 – Procesamiento de listas con car y cdr

Escribe una función '(procesa-lista lista)' que:

1. Si la lista está vacía → devuelve "Lista vacía".
2. Si el primer elemento ('car') es un número mayor a 50 → devuelve "Grande".
3. Si el primer elemento es una sublista → devuelve "Sublista detectada".
4. En cualquier otro caso → devuelve "Caso general".

Ejemplos:

```
(procesa-lista '())           ;; => "Lista vacía"  
(procesa-lista '(60 1 2))    ;; => "Grande"  
(procesa-lista '((1 2) 3 4)) ;; => "Sublista detectada"  
(procesa-lista '(10 20 30))  ;; => "Caso general"
```