ALGORITMOS DE BÚSQUEDA

1) Por profundidad:

Si lo encuentra devuelve 1 y si no 0.

```
(defun profundidad (arbol e)
        (if (null arbol)
          0
         (cond
          ((atom arbol) (if (equal arbol e) 1 0))
          ((listp arbol) (+ (profundidad (car arbol) e) (profundidad (cadr arbol) e)
(profundidad (caddr arbol) e)))
)))
```

```
CL-USER 1 > (defun profundidad (arbol e)
                    (if (null arbol)
                        0
                      (cond
                       ((atom arbol) (if (equal arbol e) 1 0))
                       ((listp arbol) (+ (profundidad (car arbol) e) (profundidad (cadr arbol) e) (profundidad (caddr arbol) e)))
)))
PROFUNDIDAD

CL-USER 2 > setq arbol '(1 (2 3) (4 (5 6) 7))
(1 (2 3) (4 (5 6) 7))

CL-USER 3 > (profundidad arbol '2)
1

CL-USER 4 > (profundidad arbol '8)
0
```

2) Por Amplitud:

- Si lo encuentra muestra el recorrido del árbol hasta encontrar el elemento deseado.
- Si no lo encuentra muestra el recorrido del árbol en su totalidad.

```
(defun amplitud (lista e)
        (when (not (null lista))
            (if (not (null (first lista)))
                (cond
                    ((atom (first lista)) (print (first lista))
                        (if (equal (first lista) e)
                            "encontrado"
                            (amplitud (cdr lista) e)))
                    ((listp (first lista)) (print (first (first lista)))
                        (if (equal (first (first lista)) e)
                            "encontrado"
                            (amplitud (append (cdr lista) (list (second (first lista))) (list
                (third (first lista)))) e)))
                    )
                (amplitud (cdr lista) e))
))
```

```
CL-USER 10 : 2 > (defun amplitud (lista e)
                   (when (not (null lista))
                    (if (not (null (first lista)))
                     (cond
                      ((atom (first lista)) (print (first lista))
                          (if (equal (first lista) e)
                              "encontrado"
                              (amplitud (cdr lista) e)))
                      ((listp (first lista)) (print (first (first lista)))
                          (if (equal (first (first lista)) e)
                             "encontrado"
                             (amplitud (append (cdr lista) (list (second (first lista))) (list (third (first lista)))) e)))
                      )
                      (amplitud (cdr lista) e))
                     ))
AMPLITUD

CL-USER 11 : 2 > (amplitud arbol '2)

1
2
"encontrado"

CL-USER 12 : 2 > (amplitud arbol '8)

1
2
4
3
5
7
6
NIL
```