

001010101010101010100101

Tipos de datos: Significado a esta información. int (32 bits), long (64 bits), char (8 bits), string (varios de 8 bits)

Implementación: COmo trabajamos estos datos y trabaja directamente con la información maquina (no es visible)

Interfaz: Las operaciones y la representación que nos ofrecen en el lenguaje de programación.

(+ 5 3)

(succ (succ (succ 8))) = 8
↑ ↑ ↑
3

(+ 8 10)

(succ (succ (----- succ 8)))
10

10 3

(succ (suma a (pred b)))

(succ (suma 10 (pred 3)))

(succ (suma 10 2))

(succ (succ (suma 10 1)))

(succ (succ (succ (suma 10 0))))

(succ (succ (succ 10)))
n

12

13

$$\Gamma() = [0]$$

$$\text{succ}(l) = (\text{cons } *t\ l)$$

$$\text{pred}(l) = (\text{cdr } l)$$

Representación

- Constructores
- Observadores <-- Predicados y extractores -->

```

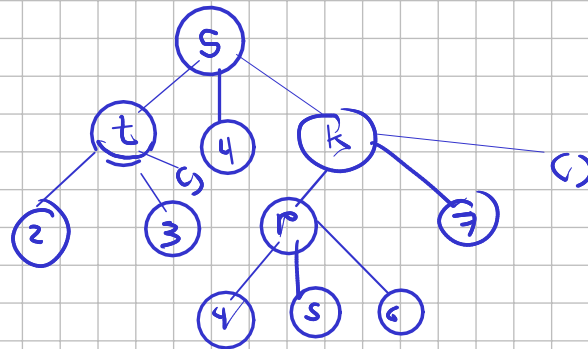
<arbol-t> ::= '()
           arbol-vacio
           ::= <numero>
           hoja(num)
           ::= <simbolo> <arbol-t> <arbol-t> <arbol-t>
           nodo(key, h1,h2,h3)

```

```

(define arbol1
  (nodo
    's
    (nodo 't (hoja 2) (hoja 3) (arbol-vacio))
    (hoja 4)
    (nodo 'k
      (nodo 'p (hoja 4) (hoja 5) (hoja 6))
      (hoja 7)
      (arbol-vacio))))

```



Ambientes

```

<enviroment> ::= '()
              (empty-env)
              ::= <list-s> <list-v> <enviroment>
              extend-env(lid lval old-env)

```

