

## Ejemplo de la utilización de recursividad en prolog con listas

```
sumarlista.pl
1 % Caso base: la suma de una lista vacía es 0
2 sumar_lista([], 0).
3
4 % Caso recursivo: suma el primer elemento y continúa con el resto
5 sumar_lista([Head|Tail], Suma) :-
6     sumar_lista(Tail, SumaTail),
7     Suma is Head + SumaTail.
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS POSTMAN CONSOLE COMMENTS

```
?- load_files('sumarlista.pl').
true.

?- make.
true.

?- make.
% /home/emafer77/Documentos/septimo-octavo/programacion-logica/listas-arboles-prolog/sumarlista compiled 0.00 sec, 0 clauses
true.

?- sumar_lista([1, 2, 3, 4], Resultado).
Resultado = 10.
```

## ejemplo de un arbol binario en prolog

```

1
2 arbol_ejemplo(
3     tree(10,
4         tree(5,
5             tree(2, nil, nil),    % Subárbol izquierdo
6             tree(7, nil, nil)    % Subárbol derecho
7         ),
8         tree(15,
9             nil,
10            tree(20, nil, nil)
11        )
12    )
13 ).
14
15
16 inorden(nil).
17
18 inorden(tree(Valor, Izquierda, Derecha)) :-
19     inorden(Izquierda),          % Recorrer subárbol izquierdo
20     write(Valor), nl,           % Escribir el valor de la raíz
21     inorden(Derecha).           % Recorrer subárbol derecho
22

```

la salida se ve de esta manera:

```

o emafer77@ASUS-TUF-Gaming-F17-FX706HM-TUF706HM-4ffcaaa9:~/Documentos/septimo-octavo/programacion-logica/listas-arboles-prolog$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [arbolbinario].
true.

?- arbol_ejemplo(Arbol), inorden(Arbol).
2
5
7
10
15
20
Arbol = tree(10, tree(5, tree(2, nil, nil), tree(7, nil, nil)), tree(15, nil, tree(20, nil, nil))).

```