

## Implementación de un verificador de buen etiquetamiento de árboles en Prolog.

### Descripción:

- Puede encontrar la descripción de la implementación aquí.
- La implementación de Prolog utilizada para el desarrollo fue swi-prolog, versión 6.6.

### Antes de utilizar alguno de los predicados:

1. Abra su intérprete de comandos y ejecute el intérprete de *swi-prolog*.  
`~$ prolog`
2. Luego de esto, proceda a cargar el archivo de la base de conocimientos que contiene los predicados.  
`?- [arista].`
3. A partir de este momento, podrá utilizar cualquiera de los predicados definidos.

### Sintaxis y semántica de los predicados:

- Si desea verificar el buen etiquetamiento de un árbol, use el predicado *bienEtiquetado(+Arbol)/1*

*% Ejemplo:*

```
bienEtiquetado(nodo(4,[arista(1,nodo(3,[])),arista(2,nodo(2,[])),arista(3,nodo(1,[]))])).
```

y el intérprete dará la respuesta correspondiente:

```
?- true.
```

- En caso de que desee generar todos los árboles *R-arios* con *N* nodos ( $N > 0$ ), use el predicado *esqueleto(+N,+R,-esqueleto)*

*% Ejemplo:*

```
esqueleto(3,2,Esqueleto).
```

y el intérprete dará la respuesta correspondiente:

```
?- esqueleto(3,5,Esqueleto).
```

```
Esqueleto = [[1], [1], [0]] ;
```

```
Esqueleto = [[2], [0, 0]] .
```

- Si desea obtener árboles bien etiquetados a partir de un esqueleto, utilice el predicado *etiquetamiento(+Esqueleto,-Arbol)*

*% Ejemplo:*

```
etiquetamiento(esq([[1], [1], [0]]),Etiquetable).
```

y el intérprete dará la respuesta correspondiente:

```
Etiquetable = nodo(1, [arista(2, nodo(3, [arista(1, nodo(2, []))]))]) ;
```

```
Etiquetable = nodo(2, [arista(1, nodo(1, [arista(2, nodo(3, []))]))]) ;
```

```
Etiquetable = nodo(2, [arista(1, nodo(3, [arista(2, nodo(1, []))]))]) ;
```

```
Etiquetable = nodo(3, [arista(2, nodo(1, [arista(1, nodo(2, []))]))]) .
```

- En caso de que se desee verificar si todos los esqueletos de árboles *R-arios* con *N nodos* son bien etiquetables, use el predicado *esqEtiquetables(+R,+N)*

*% Ejemplo:*

```
esqEtiquetables(3,5).
```

y el intérprete dará la respuesta correspondiente:

```
?- true.
```

- Si desea mostrar en pantalla un árbol en un formato más sencillo de leer, use el predicado *describirEtiquetamiento(+Arbol)*.

*% Ejemplo:*

```
describirEtiquetamiento(nodo(4,[arista(1,nodo(3,[])),arista(2,nodo(2,[])),arista(3,nodo(1,[]))])).
```

y el intérprete dará la respuesta correspondiente:

```
nodo(4
  [arista(1
    nodo(3
      []))
  arista(2
    nodo(2
      []))
  arista(3
    nodo(1
      []))
])
true
```