

Ejercicio. Dada la implementación “nodo-lista” (es decir, “primogénito–siguiente hermano”) de *tries* vista en clase, se debe desarrollar un procedimiento que escriba en pantalla en orden alfabético todas las palabras almacenadas en un *trie*.

Solución:

```
procedimiento escribe(ent t:trie)
variable palabra:cadena
```

```
procedimiento preTrie(ent t:trie; ent palabra:cadena)
{Adaptación del recorrido en pre-orden para un bosque de árboles n-arios visto
 en la lección 15. En el parámetro 'palabra' se recibe la concatenación de caracteres
 del camino que va de la raíz del trie al padre del nodo apuntado por t}
principio
  si t≠nil entonces
    preOrden(t,palabra);           {recorrido del árbol apuntado por t}
    preTrie(t↑.sigHermano,palabra) {recorrido de los demás árboles hermanos de t}
  fsi
fin
```

```
procedimiento preOrden(ent t:trie; ent palabra:cadena)
{Adaptación del recorrido en pre-orden para un árbol n-ario visto en la lección 15}
principio
  si t↑.dato='$' entonces
    escribirLínea(palabra)
  sino
    preTrie(t↑.primogenito,palabra + t↑.dato) {+ es aquí la concatenación de cadenas}
  fsi
fin
```

```
principio {de escribe}
  palabra:="";           {"" representa la cadena vacía}
  preTrie(t,palabra)
fin
```