

Árboles

Santi Seguí

Objetivos

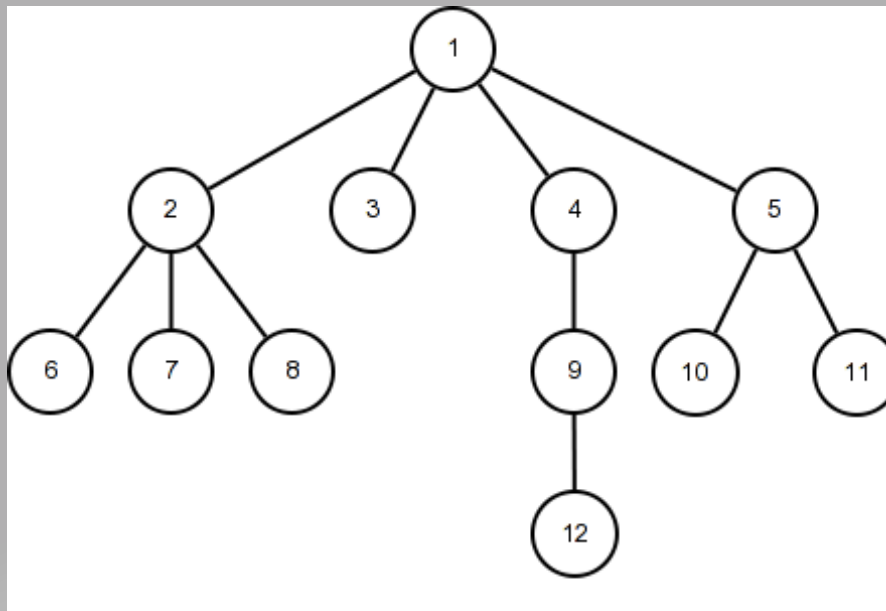
- Entender qué tipos de estructuras hay
- Entender qué es una estructura de árbol y para qué sirve
- Definir la estructura del árbol recursivamente
- ¿Qué componentes tiene un árbol?
- ¿Qué propiedades tiene un árbol?

Los arboles

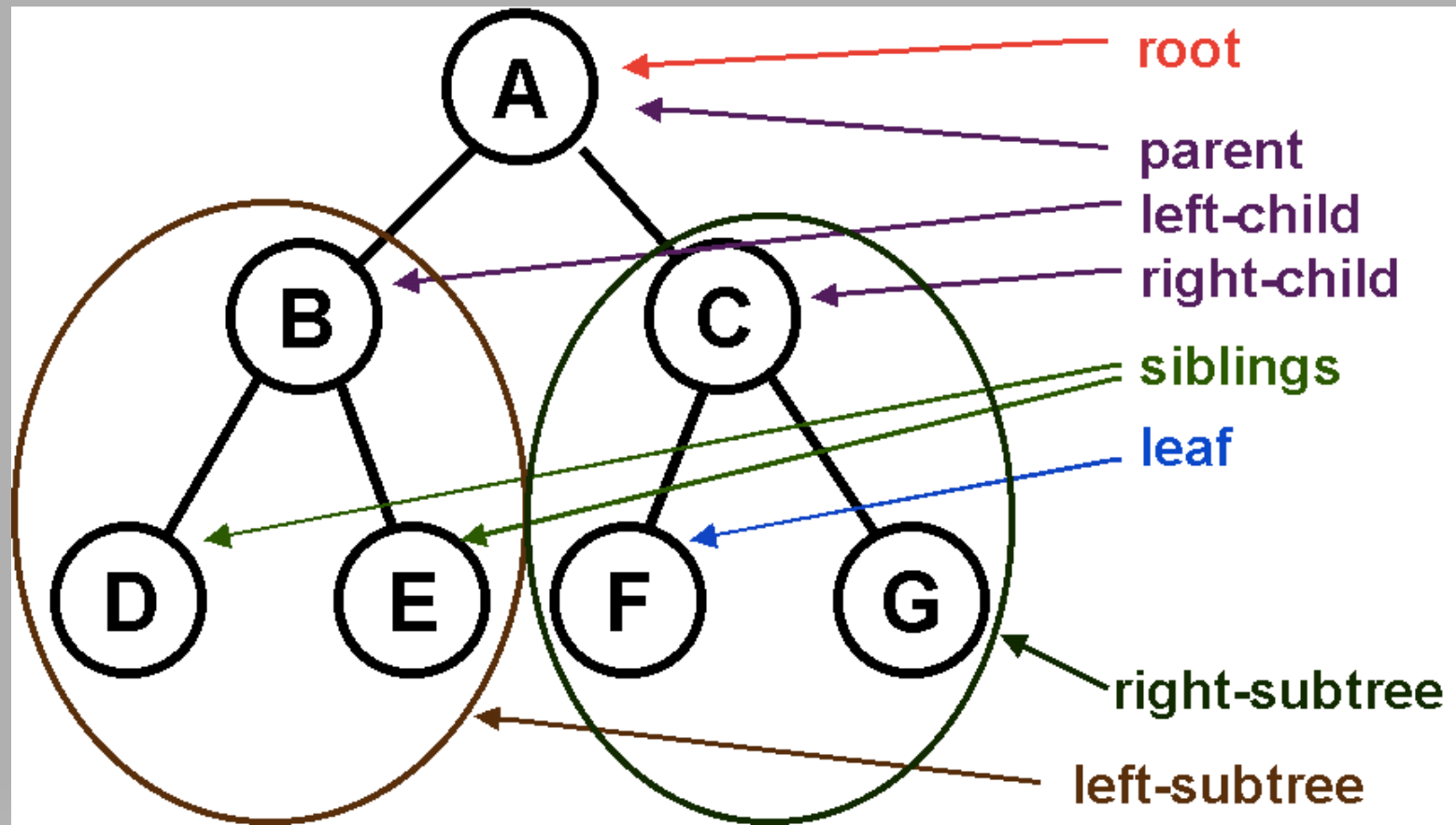
- Las listas enlazadas son estructuras lineales
 - Son secuenciales, un elemento detrás de otro
- Los árboles
 - Junto con los grafos son estructuras de datos no lineales
 - Superan las desventajas de las listas
 - Distintas formas de recorrido
- Para que pueden ser muy útiles?
 - Son **muy útiles** para la **búsqueda y recuperación de información**

Colecciones jerárquicas

- Estructura árbol

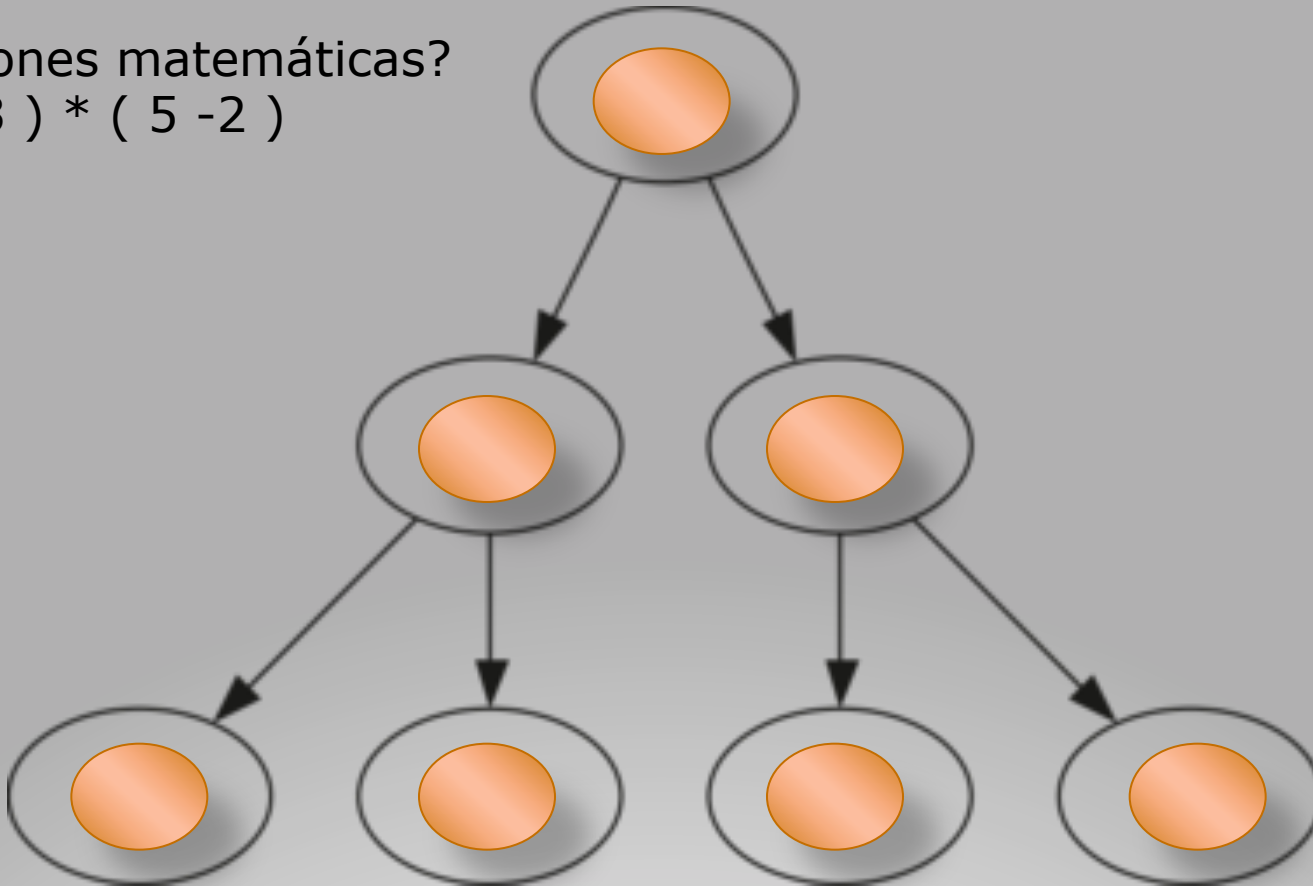


Elementos de los Árboles

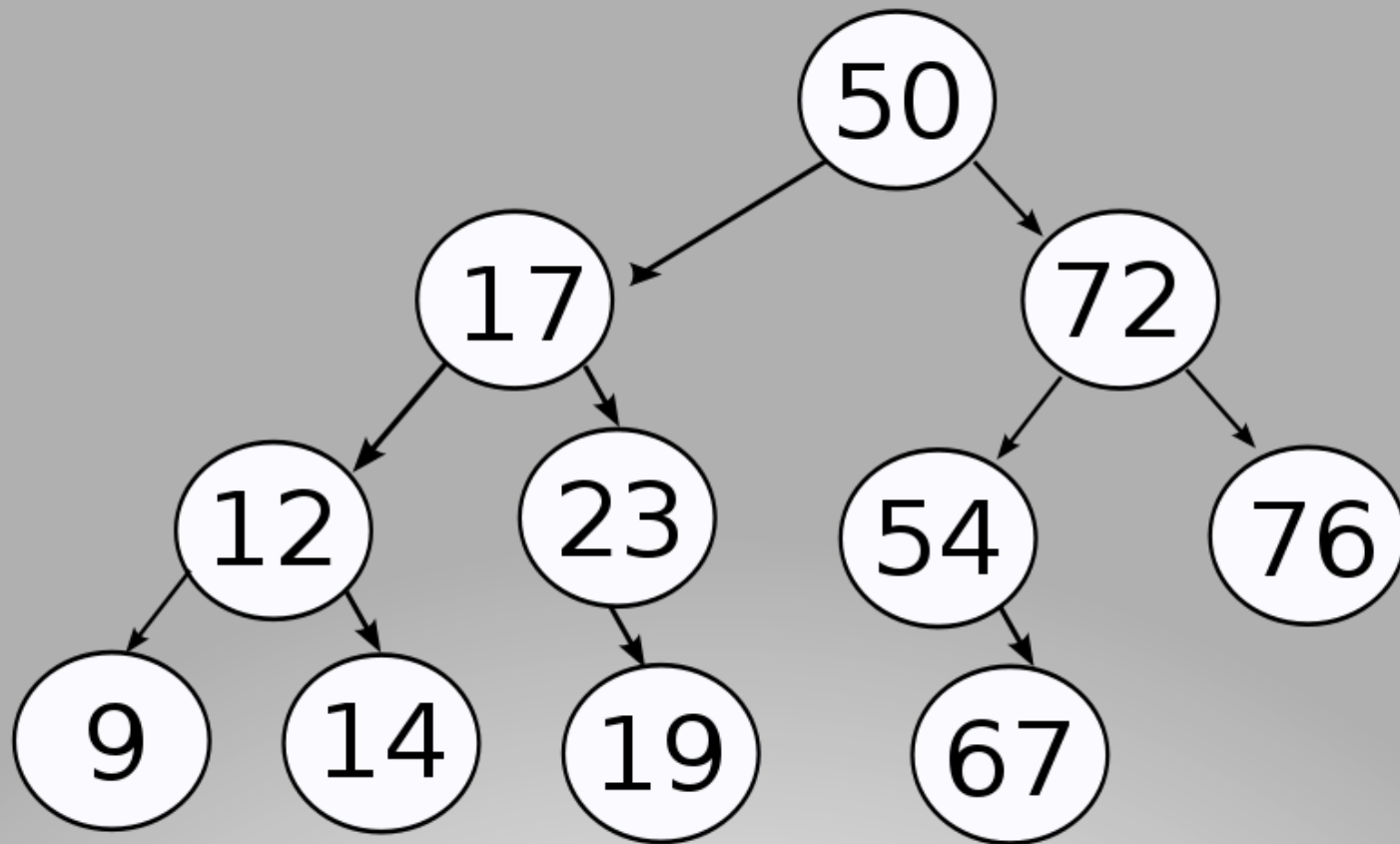


Ejemplos

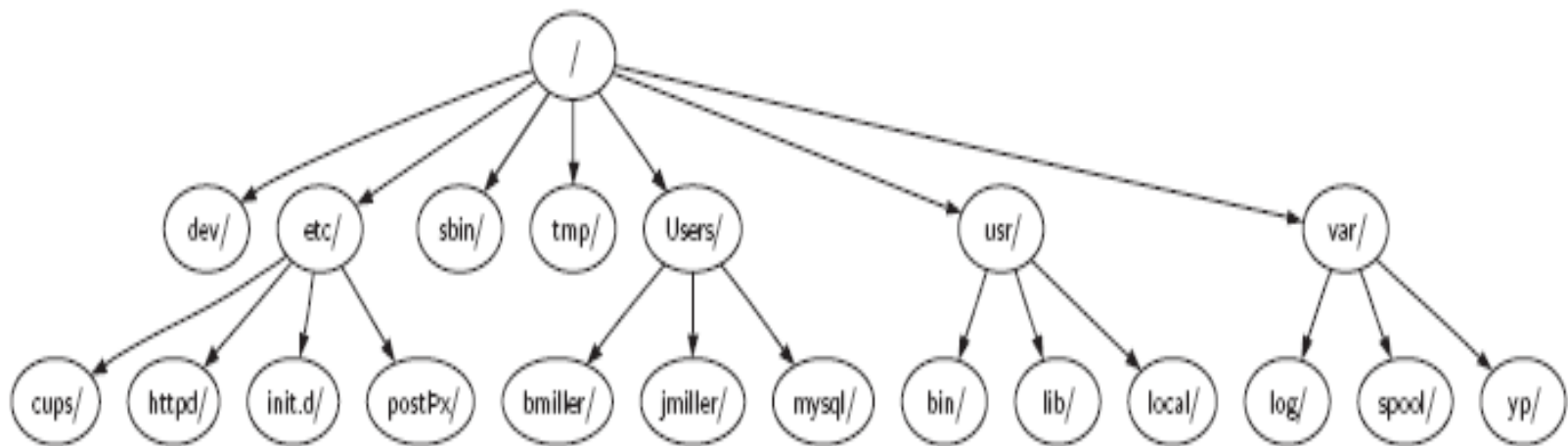
Ecuaciones matemáticas?
 $(7 + 3) * (5 - 2)$



Ejemplos

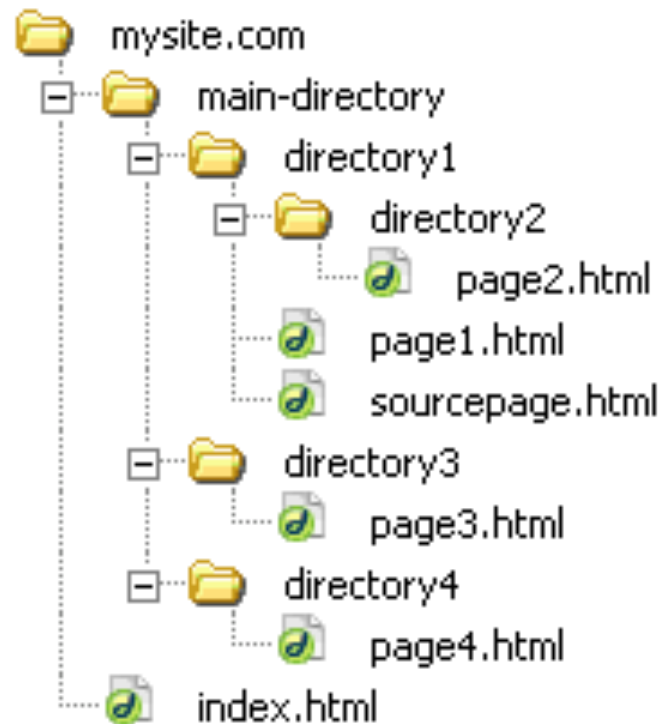


La jerarquía del sistema de ficheros en Unix

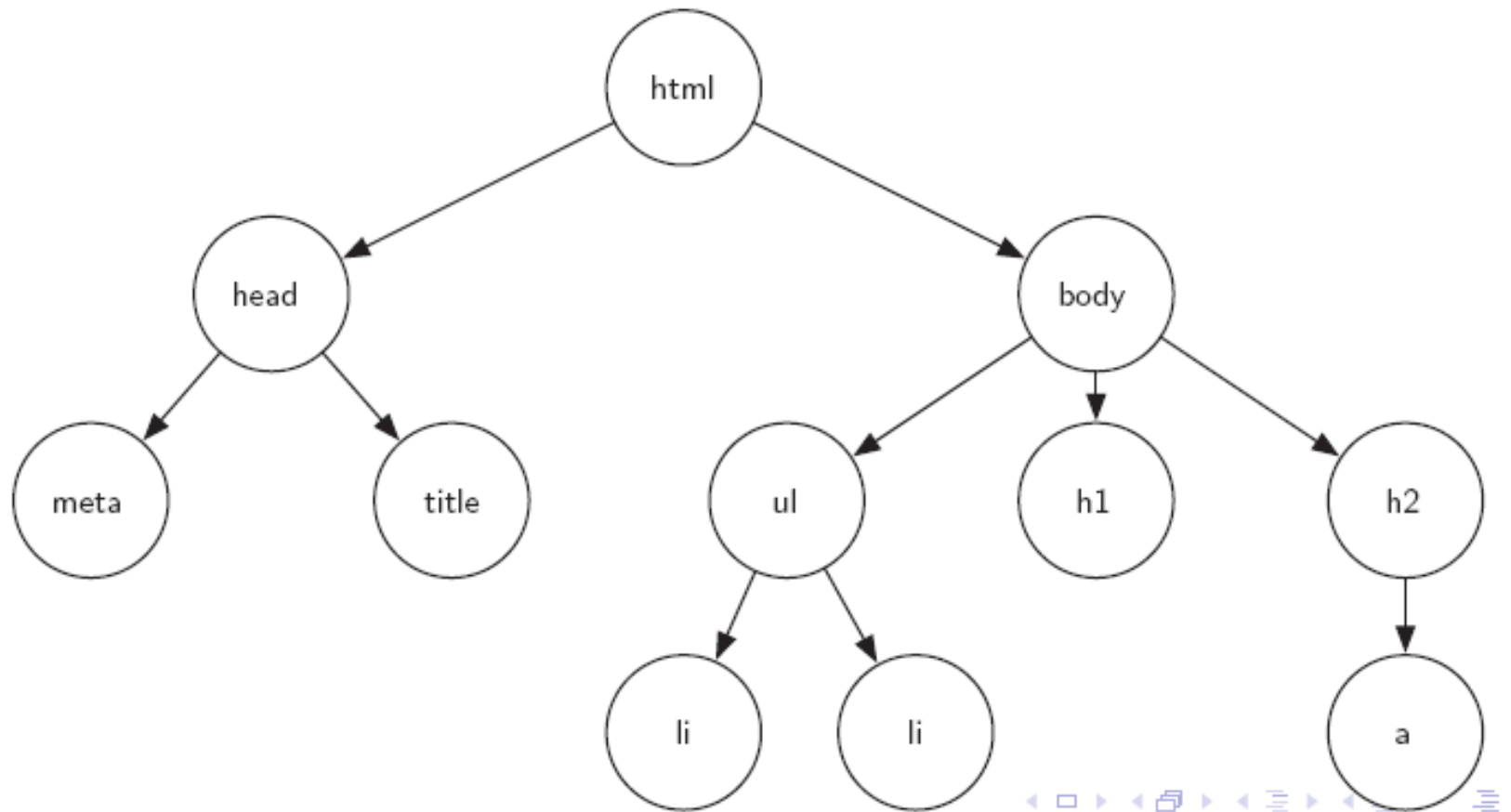


Estructura de directorios y ficheros

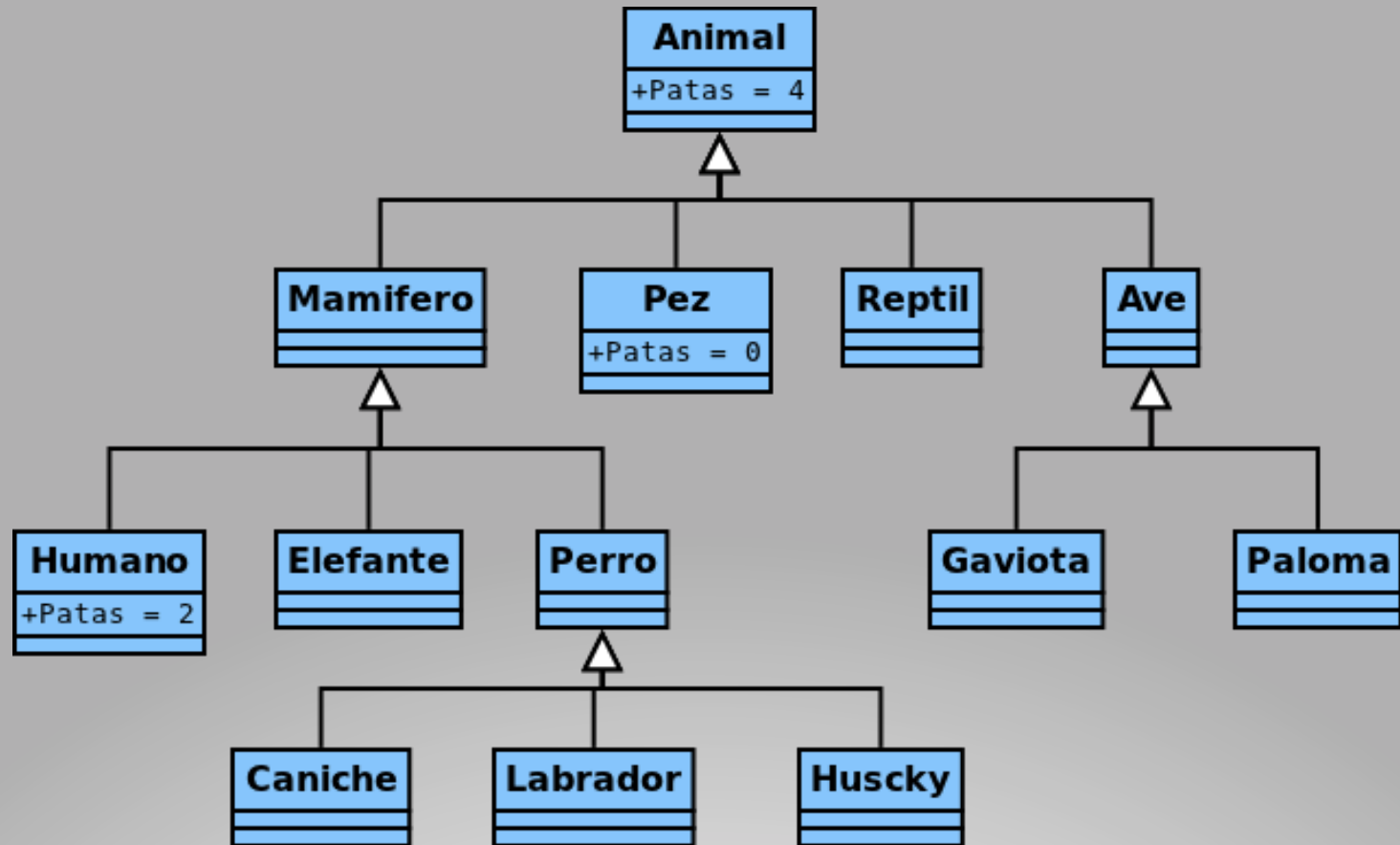
Example of a Simple Directory Structure



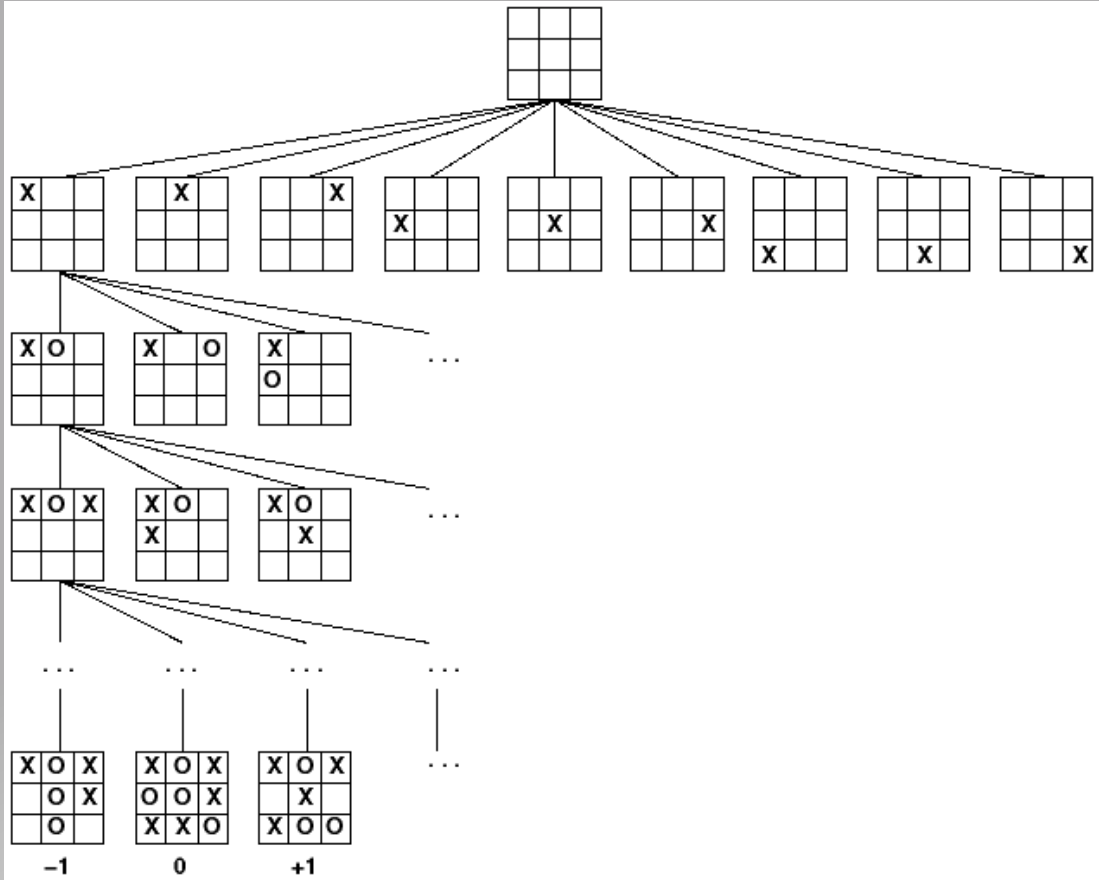
Árbol correspondiente a las etiquetas de una página web



Ejemplos

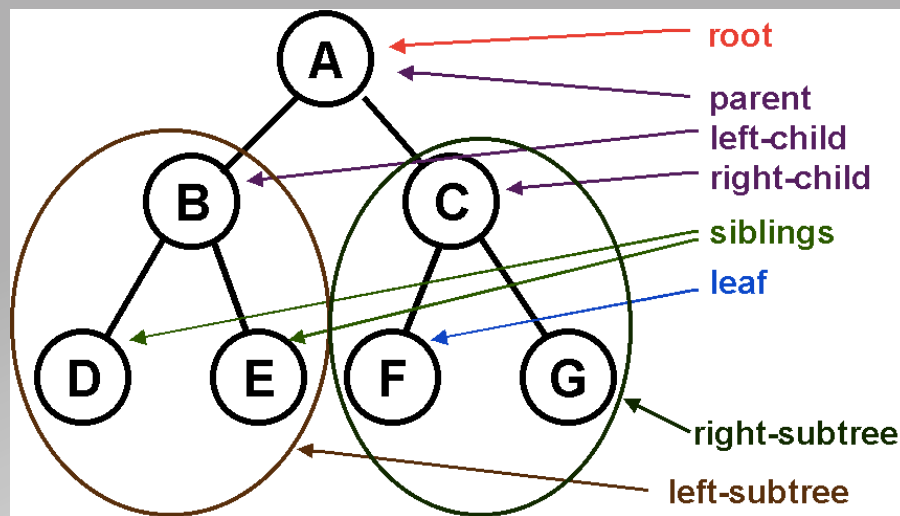


Ejemplos



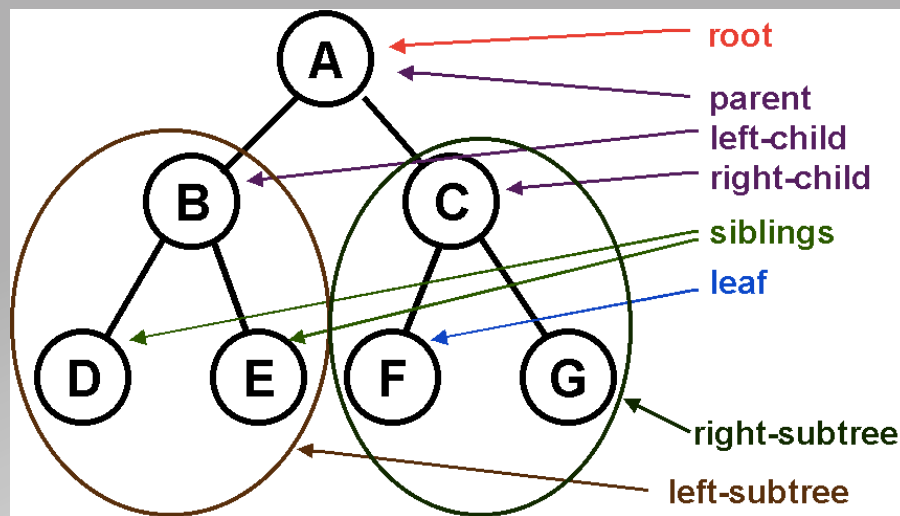
¿Qué componentes tiene un árbol?

- El árbol está compuesto por **nodos** y **ramas** que **conectan los nodos**, de forma que **cada nodo contiene un único predecesor!**
- El **nodo** contiene la información
- Las **ramas** la conexión a otros nodos



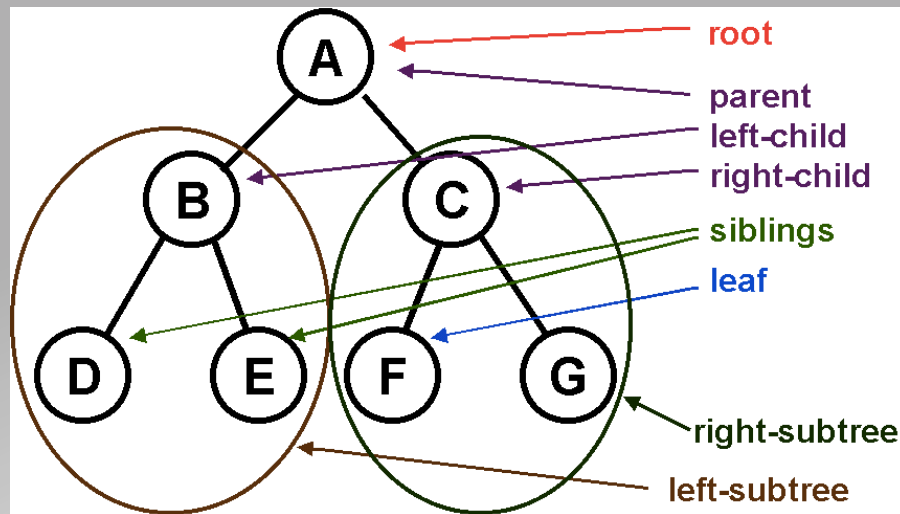
¿Qué componentes tiene un árbol?

- El **nodo raíz** (root) es el nodo principal (no hay ninguna rama que llega a este nodo)
- **Nodo padre de un nodo n** es el nodo desde donde sale la única rama que entra en el nodo n.
- **Nodo hoja**: un nodo que no tiene hijos



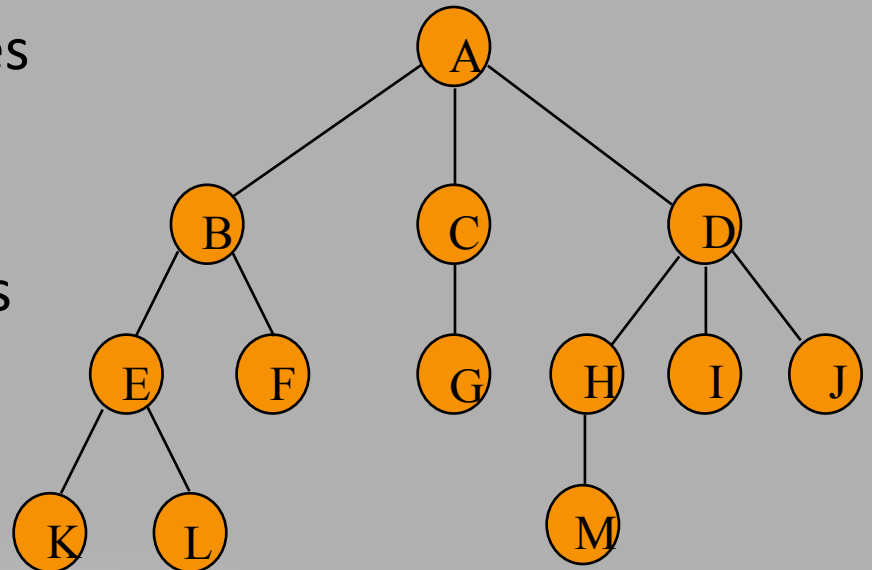
¿Qué componentes tiene un árbol?

- **Nodos hermanos:** son los nodos que tienen el mismo padre
 - El nodo raíz es el ancestro de todos los nodos
- **Ancestros:** son el padre de un nodo o el padre de algún nodo ancestro.
 - El nodo raíz es el ancestro de todos los nodos
- **Nodo descendiente:** el hijo de un nodo o el hijo de un descendiente suyo.
 - Todos los nodos son descendientes del nodo raíz.

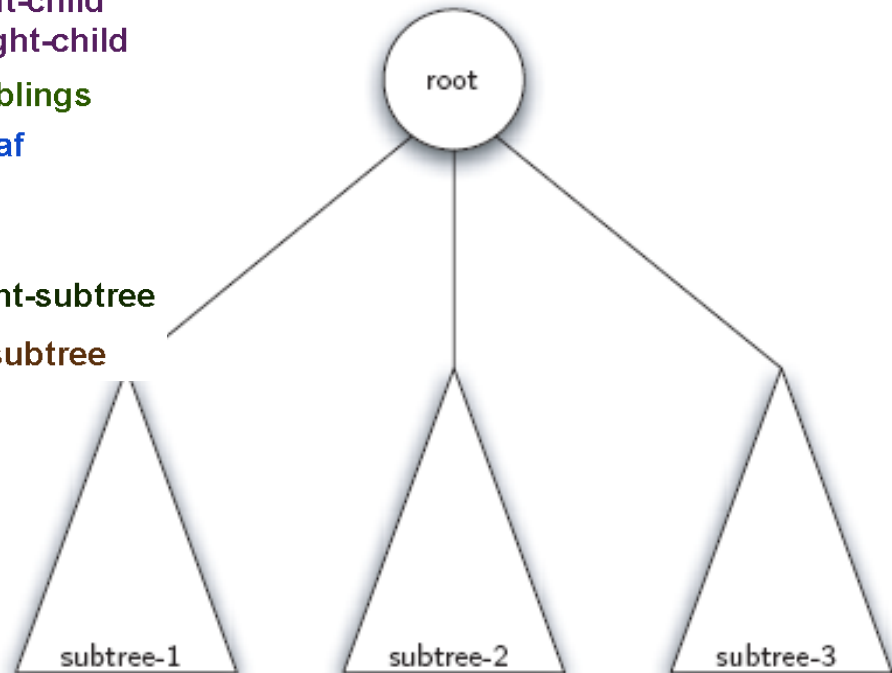
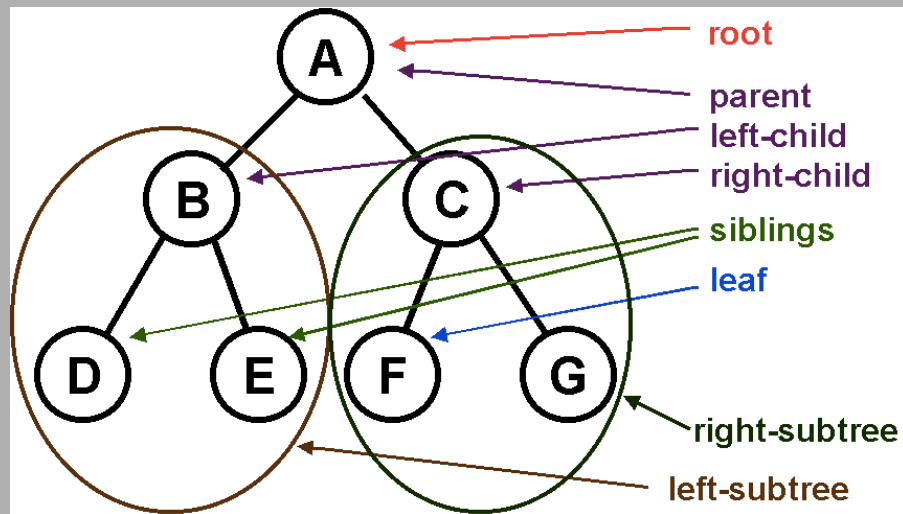


Propiedades de un árbol

- Primera propiedad: los árboles son jerárquicos
- Segunda propiedad: Todos los hijos de un nodo son independientes
- Tercera propiedad: El camino hasta cualquier hoja es único



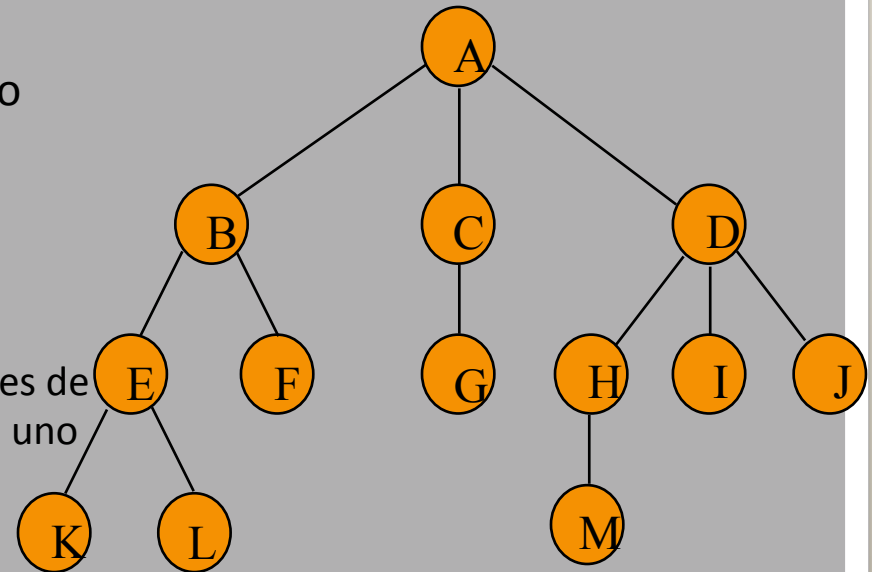
Definición recursiva de un árbol



¿Qué componentes tiene un árbol?

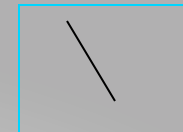
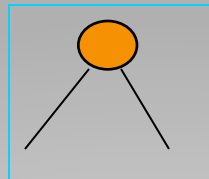
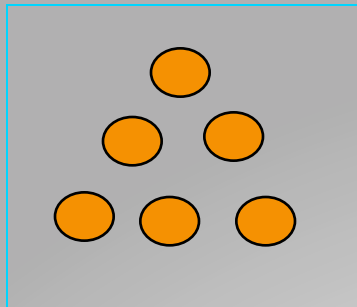
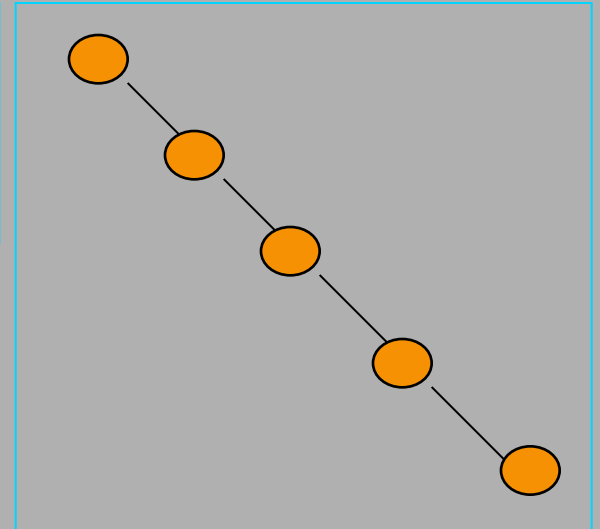
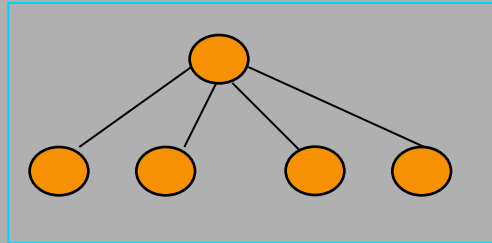
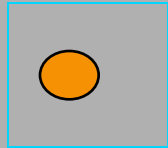
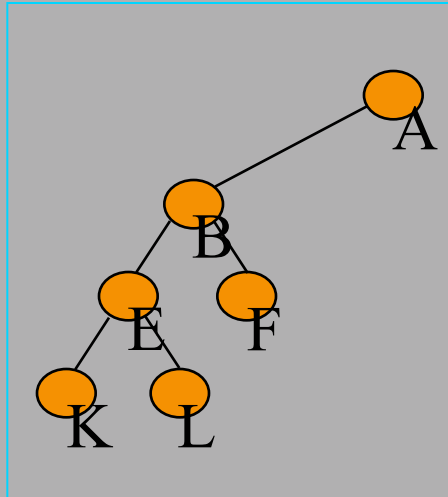
- Definición recursiva: el árbol es un conjunto finito de nodos que cumple

- Existe un nodo especial llamado raíz
- El resto de nodos están en n ($n \geq 0$) particiones de conjuntos disjuntos T_1, T_2, \dots, T_n donde cada uno de estos conjuntos es un árbol.
 - T_1, T_2, \dots, T_n son los subárboles del nodo raíz.



- Los subárboles de un nodo son disjuntos e.d. no puede haber conexiones entre los subárboles de un nodo.
- Nota: es una definición recursiva. Facilita la definición y la aplicación de funciones recursivas.

¿ Son árboles?

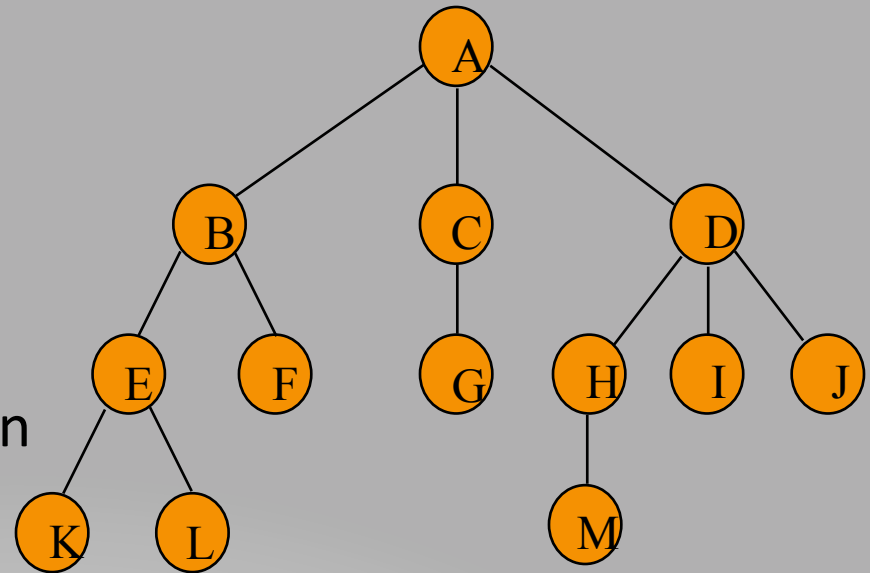


Terminología de los Árboles

- Nivel de un Nodo
- Profundidad de un Nodo
- Altura (o profundidad) de un Árbol
- Altura de un Nodo
- Grado de un Árbol – Grado de un Nodo
- Logitud de camino de un Árbol

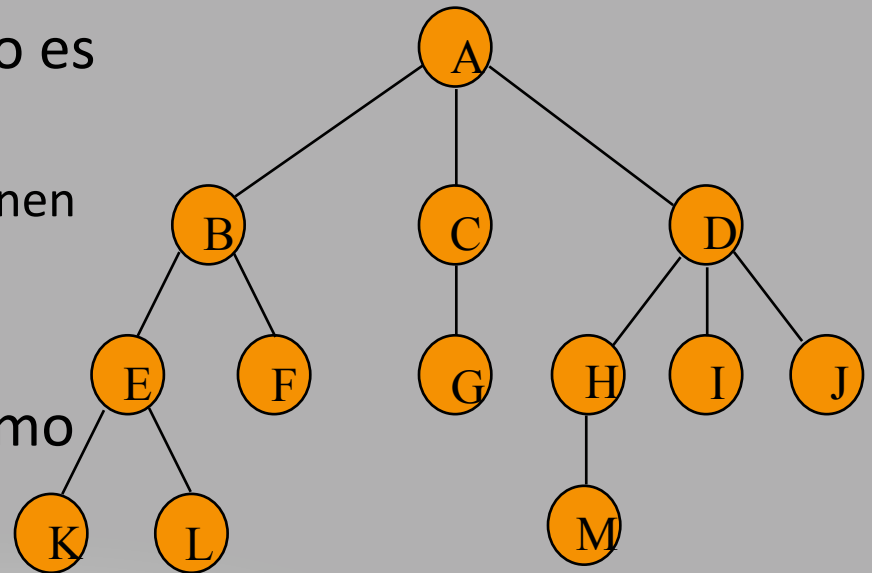
¿Nivel de un nodo?

- El **nivel** de un nodo se define como sigue:
 - El nivel del nodo raíz es 1
 - Si un nodo está en el nivel L , sus hijos están en el nivel $L+1$
- La **altura** o **profundidad** de un árbol es el máximo nivel

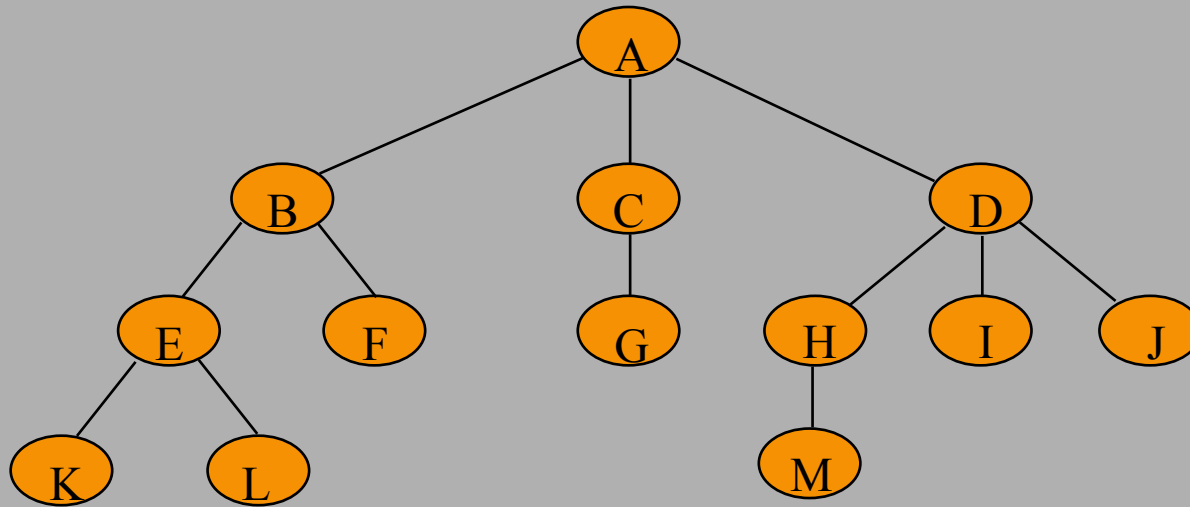


¿Cómo medimos el grado de un árbol?

- El número de hijos de un nodo es el **grado del nodo**
 - Las **hojas o nodos terminales** tienen **grado cero**
- **Grado** de un árbol: es el máximo grado de sus nodos



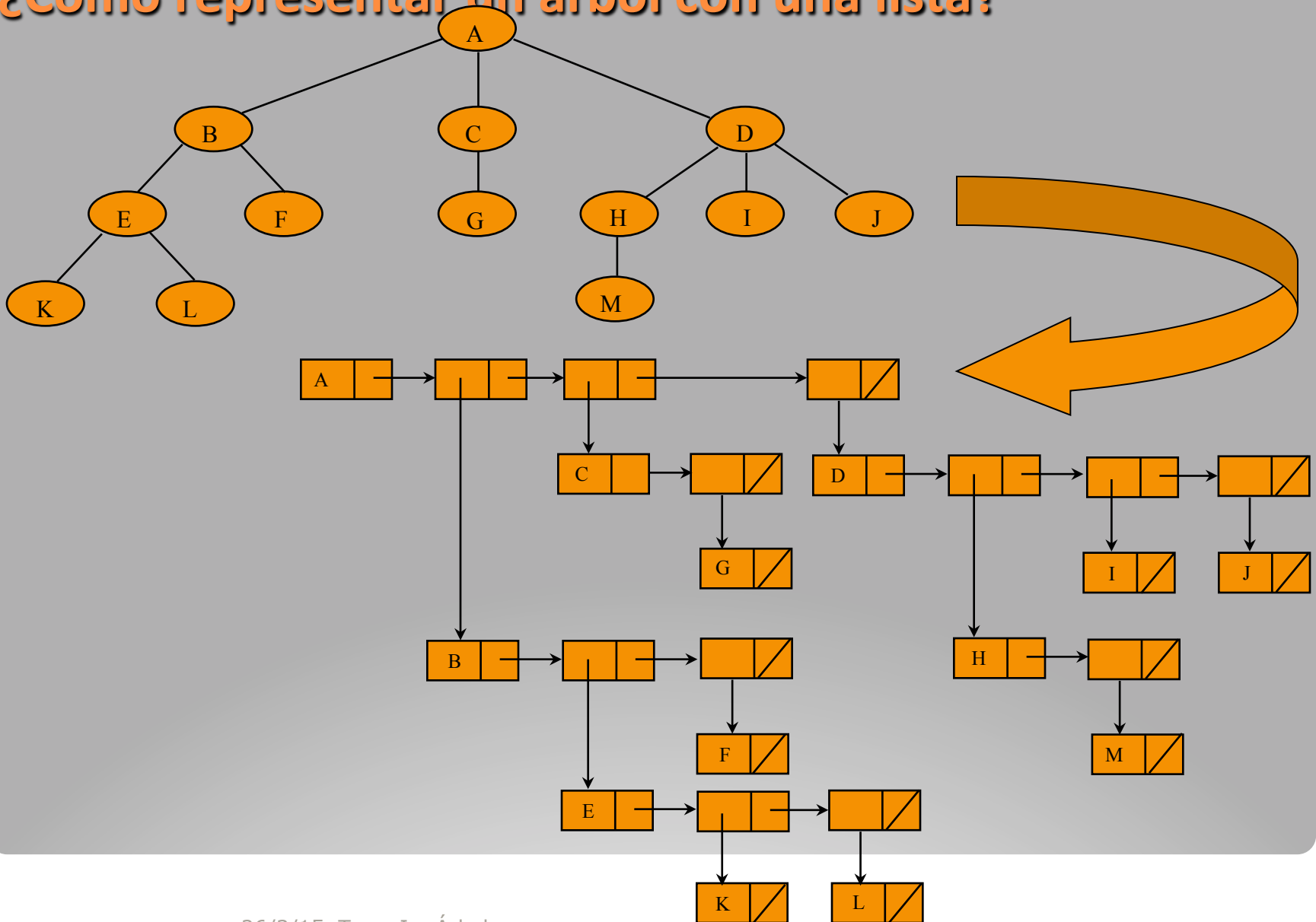
¿Cómo representar un árbol?



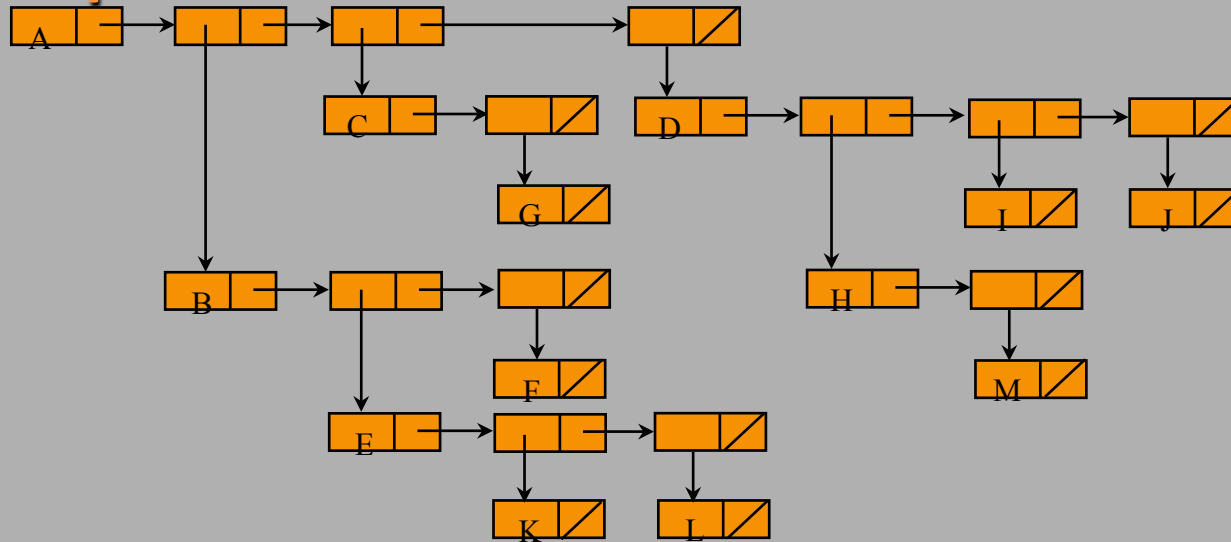
- Podemos representar un árbol como una lista enlaza:

$(A(B(E(K,L),F),C(G),D(H(M),I,J)))$

¿Cómo representar un árbol con una lista?



¿Cómo representar un árbol con una lista?



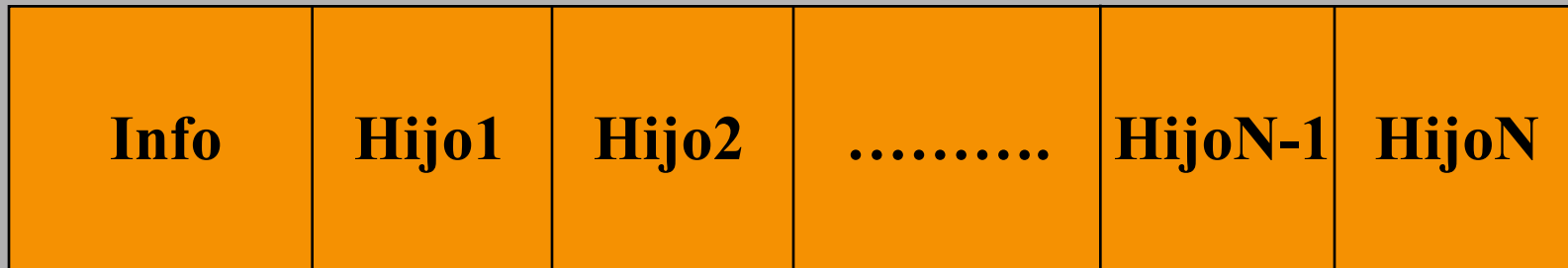
¿Si un nodo tiene n hijos, cuántos nodos auxiliares necesita?

=> Se utilizan mucho más nodos de los que tiene el árbol (desperdicio de memoria).

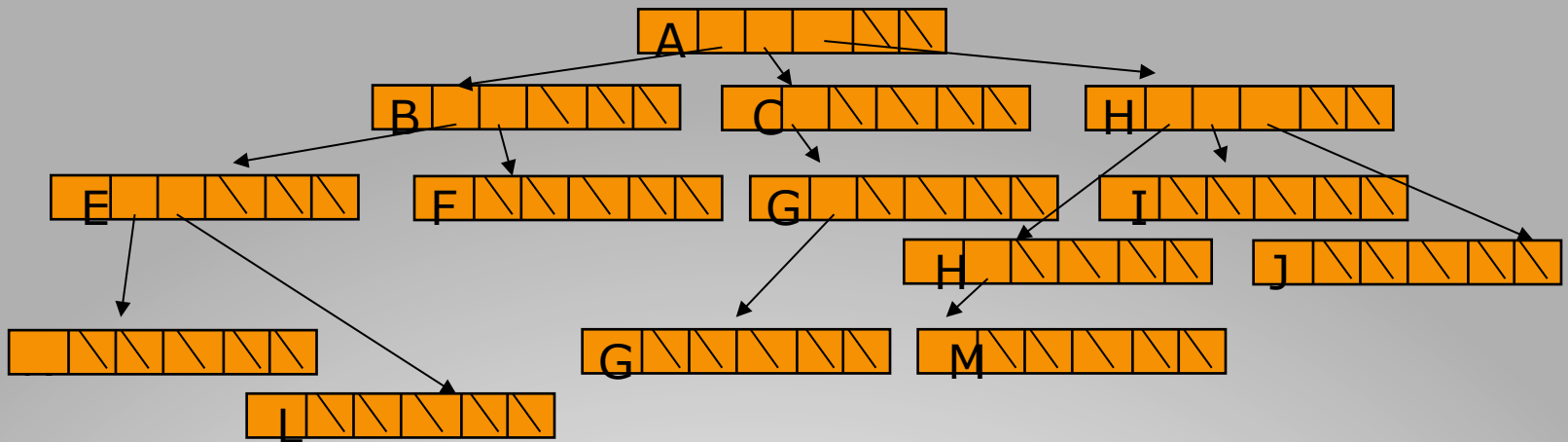
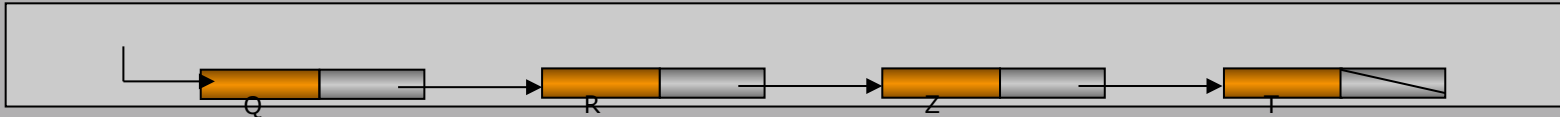
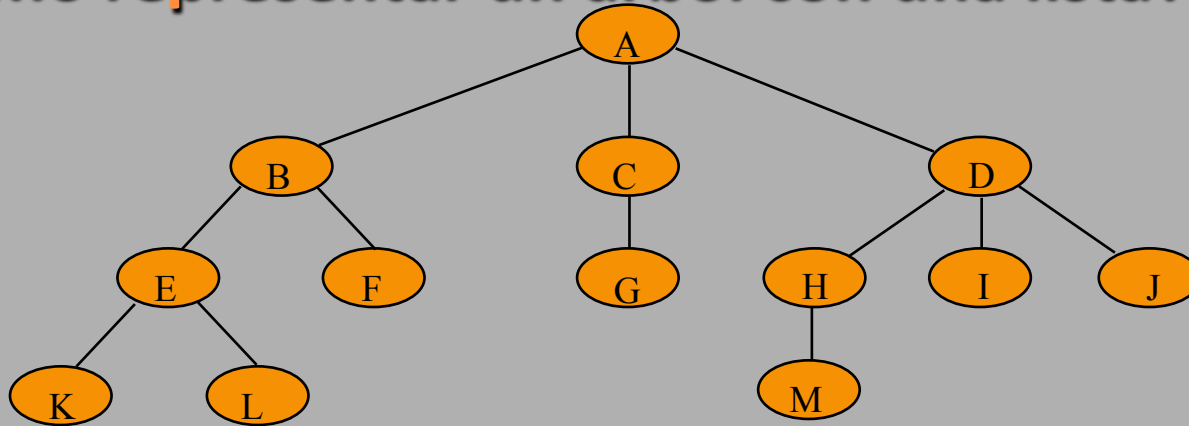
=> Se dificulta el diseño de los algoritmos

¿Cómo representar un árbol con una lista?

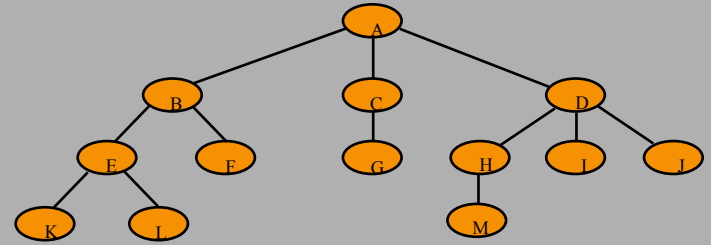
- Es deseable tener una representación específica para árboles
- El tamaño de los nodos (en particular, el número de enlaces) puede ser variable o fijo
- Se suele utilizar tamaño fijo según el grado del árbol



¿Cómo representar un árbol con una lista?



¿Cuál es el grado óptimo de los árboles?



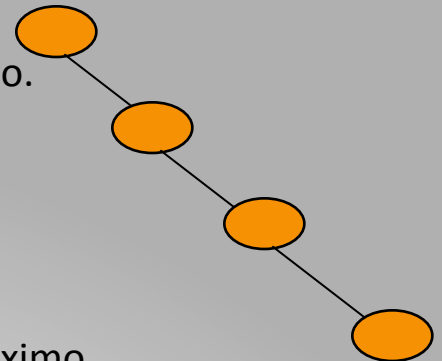
Propiedad: Si T es un árbol de grado k con n nodos de tamaño fijo, entonces: $n*(k-1)+1$ de los enlaces son None, $n \geq 1$.

Demostración:

- El número de campos hijos distintos de cero en un árbol de n nodos es
 - $n-1$ (cada nodo es referenciado por una conexión excepto el nodo raíz).
- El número total de campos hijo en un árbol de grado k con n nodos será:
 - $n*k$.
- Los enlaces vacíos son el complemento de los enlaces ocupados:
 - $(n*k)-(n-1)=n*(k-1)+1$.

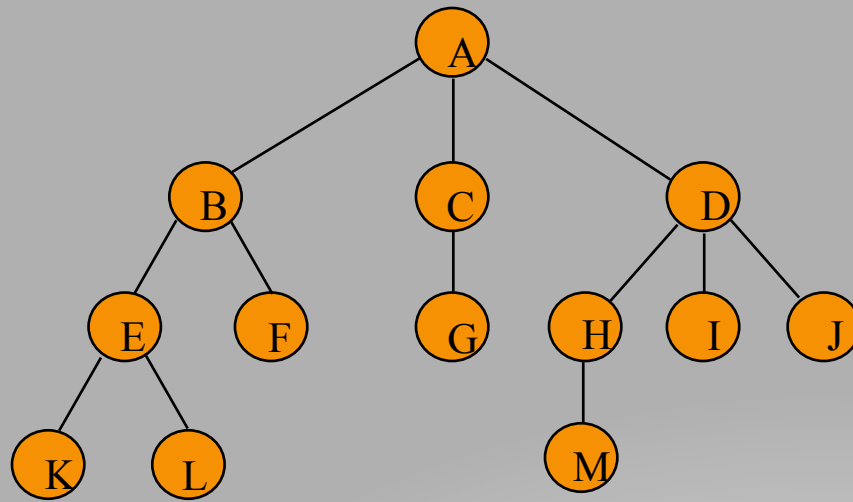
¿Cuál es el grado óptimo de los árboles?

- Los enlaces vacíos son: $(n * k) - (n - 1) = n * (k - 1) + 1$
- Nota: Imaginemos un árbol de 1.000.000 nodos y grado 2. Tendremos 1.000.000+1 enlaces vacíos.
- Imaginemos un árbol de 1.000.000 nodos y grado 10000. Tendremos 1.000.000*9999+1 enlaces vacíos.
- Teniendo la misma cantidad de información (el mismo numero de nodos), “malgastaremos” mucha memoria
- **Conclusión:** hemos de intentar trabajar con árboles de grado pequeño.
- ¿Qué grados tienen sentido?
 - Si $k=1$: tenemos una lista.
 - Si $k=2$: tenemos un árbol donde cada nodo puede tener como máximo 2 hijos. Se llama árbol binario.



¿Podemos decidir el grado del árbol?

Propiedad: Un árbol de cualquier grado se puede representar como árbol binario.



Regla: Los hijos izquierdos continúan siendo hijos izquierdos y sus hermanos pasan a ser hijos derechos.

