

# Introducción a Árboles

Estructuras de Datos

Andrea Rueda

Pontificia Universidad Javeriana  
Departamento de Ingeniería de Sistemas

# Introducción a Árboles

# Estructuras hasta ahora

- Estructuras lineales:
  - Listas, serie de elementos encadenados (contiguos).
  - Pilas, serie de elementos con acceso / inserción / eliminación sólo por el tope.
  - Colas, serie de elementos con acceso / eliminación por cabeza, inserción por cola.
- Relaciones de mayor-menor, reciente-antiguo, primero-último, anterior-siguiente, ...
- Recorridos de derecha-izquierda, izquierda-derecha.

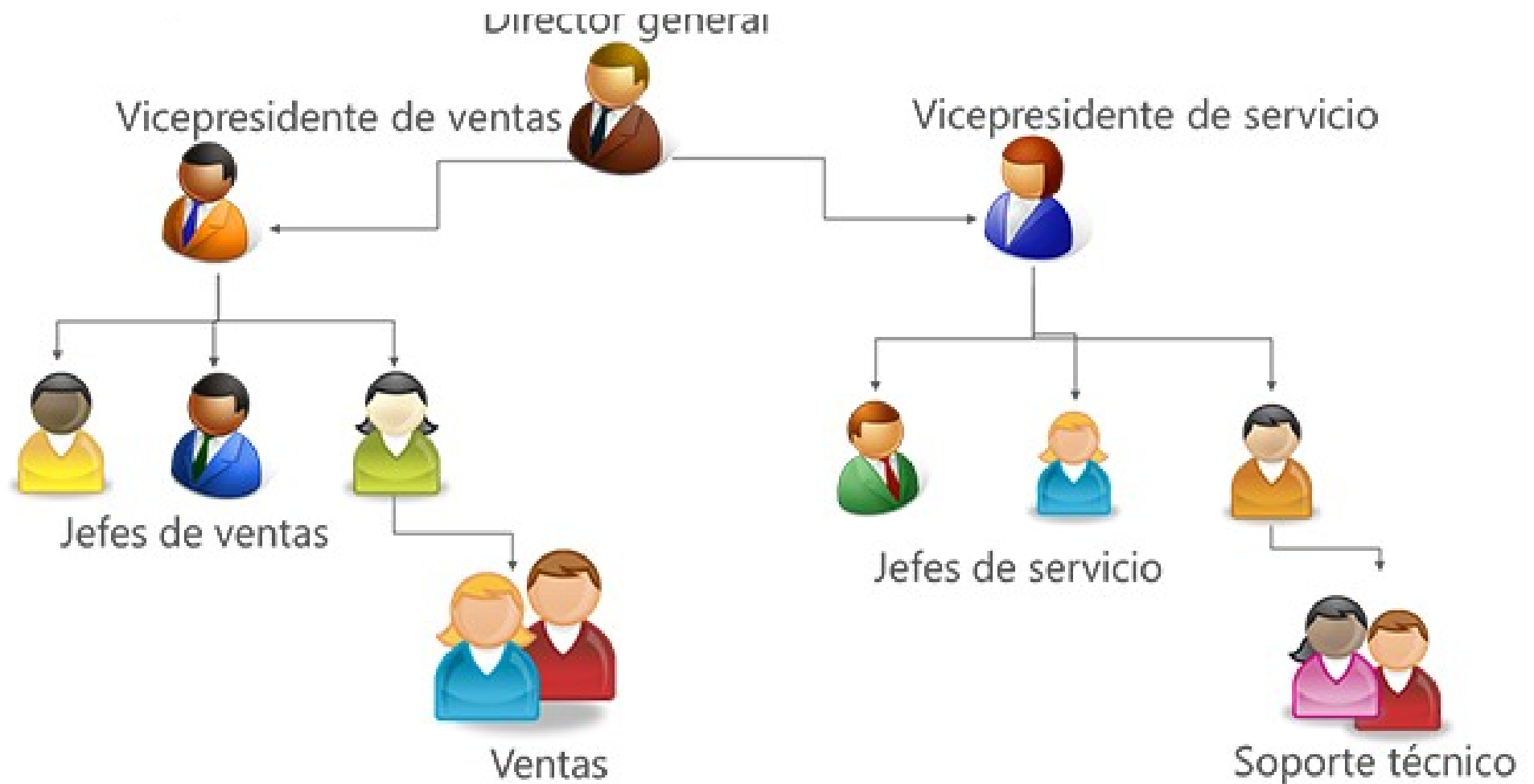
# Estructuras hasta ahora

- ¿Cómo podríamos modelar otro tipo de relaciones?
  - ▶ Contenencia (ej: gato, león, tigre pertenecen al género animal: felinos).
    - ¿Composición? ¿Multilista?
  - ▶ Jerarquía (ej: estructura de cargos en una empresa).
    - ¿?

# Jerarquías

- Árbol genealógico.
- Tablas de contenidos.
- Sistemas de archivos.
- Taxonomías.
- Redes de computadores.
- ...

# Jerarquías



# Árboles

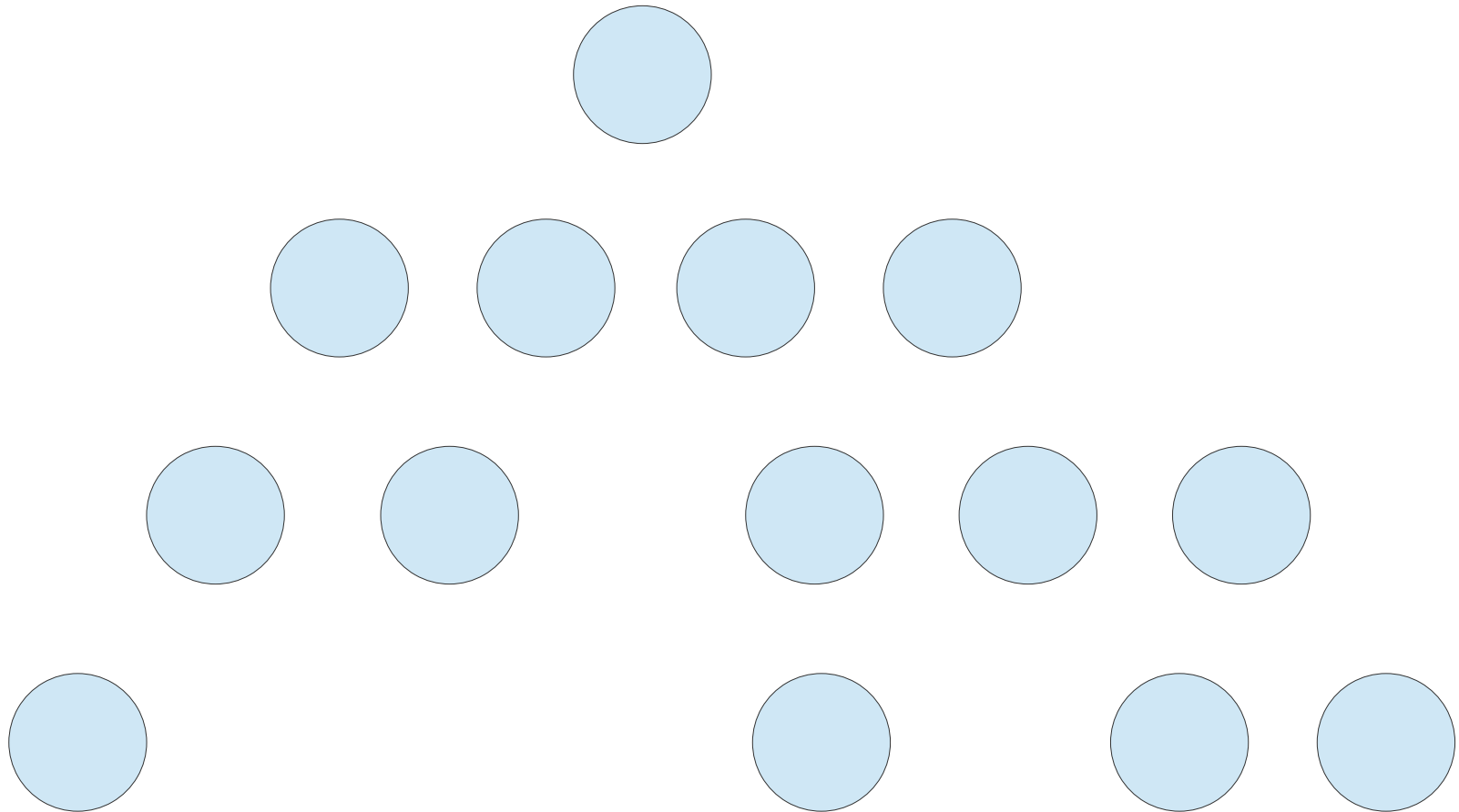
- Estructura no-lineal usada para representar entidades relacionadas por medio de jerarquías.
- Contenedor “óptimo” recurrente.
- Aplicaciones:
  - Bases de datos → indexamiento de información.
  - Juegos → descripción de posibles jugadas.
  - Simuladores.
  - Optimización/búsqueda numérica.
  - Representación de expresiones aritméticas.

# Árboles

- Definición: conjunto finito de elementos del mismo tipo, conectados por un conjunto finito de líneas dirigidas.
  - Elementos: nodos.
  - Líneas dirigidas: ramas.
- Conjunto vacío: árbol vacío.
- Conjunto unitario: único nodo  $\rightarrow$  raíz.
- Conjunto de dos o más: raíz + conjuntos disyuntos de nodos  $\rightarrow$  cada uno un (sub)árbol.

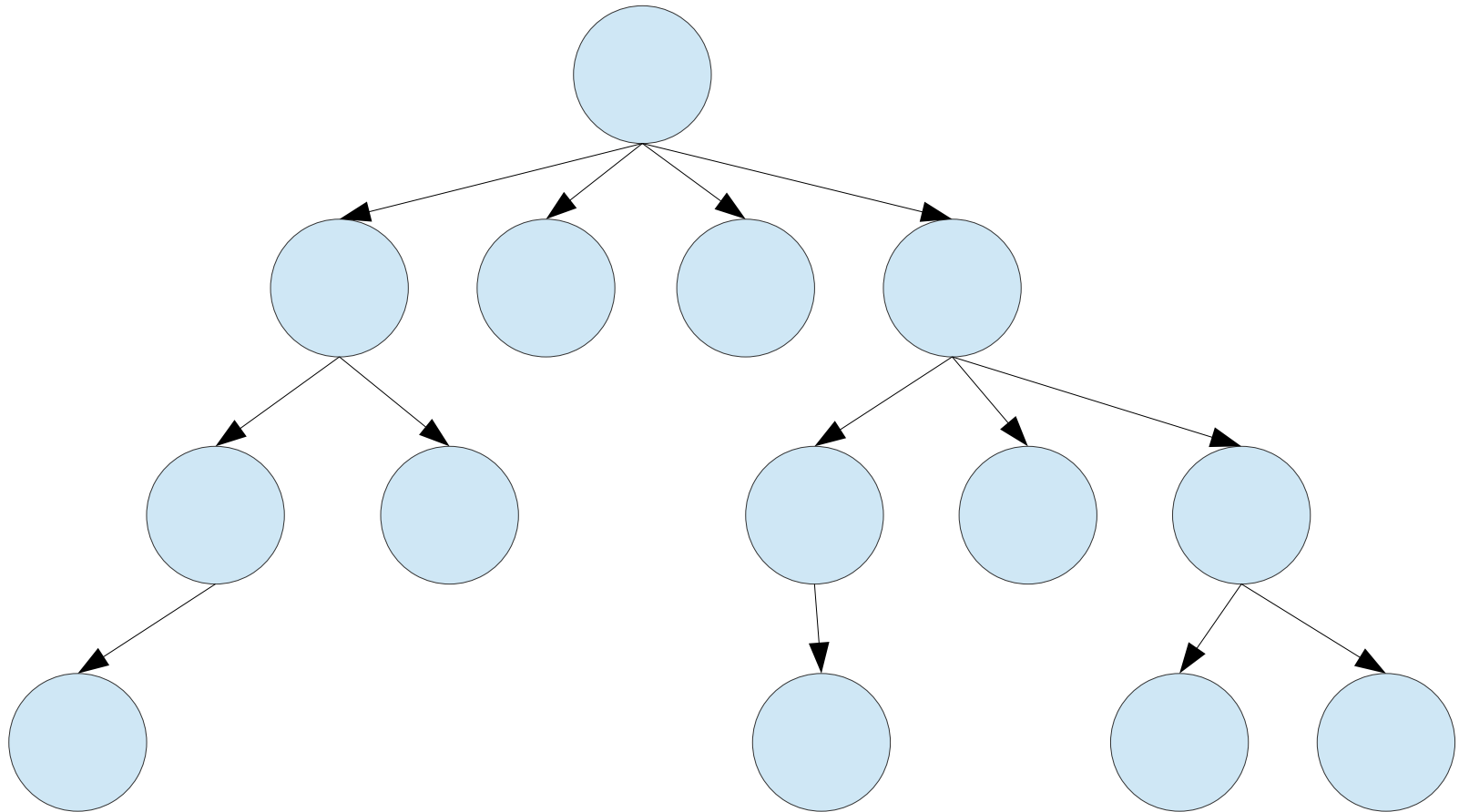


# Árboles



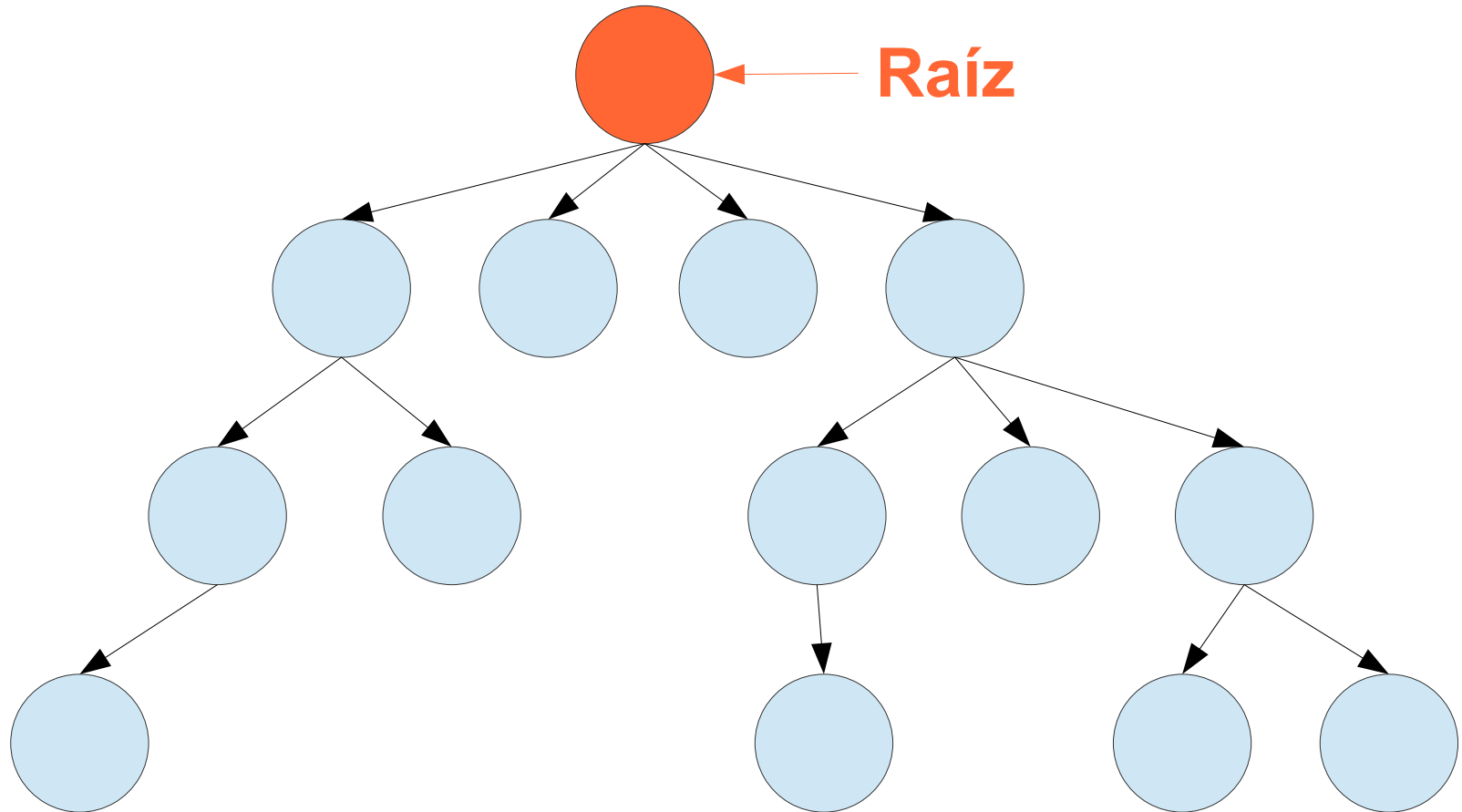
**Nodos**

# Árboles

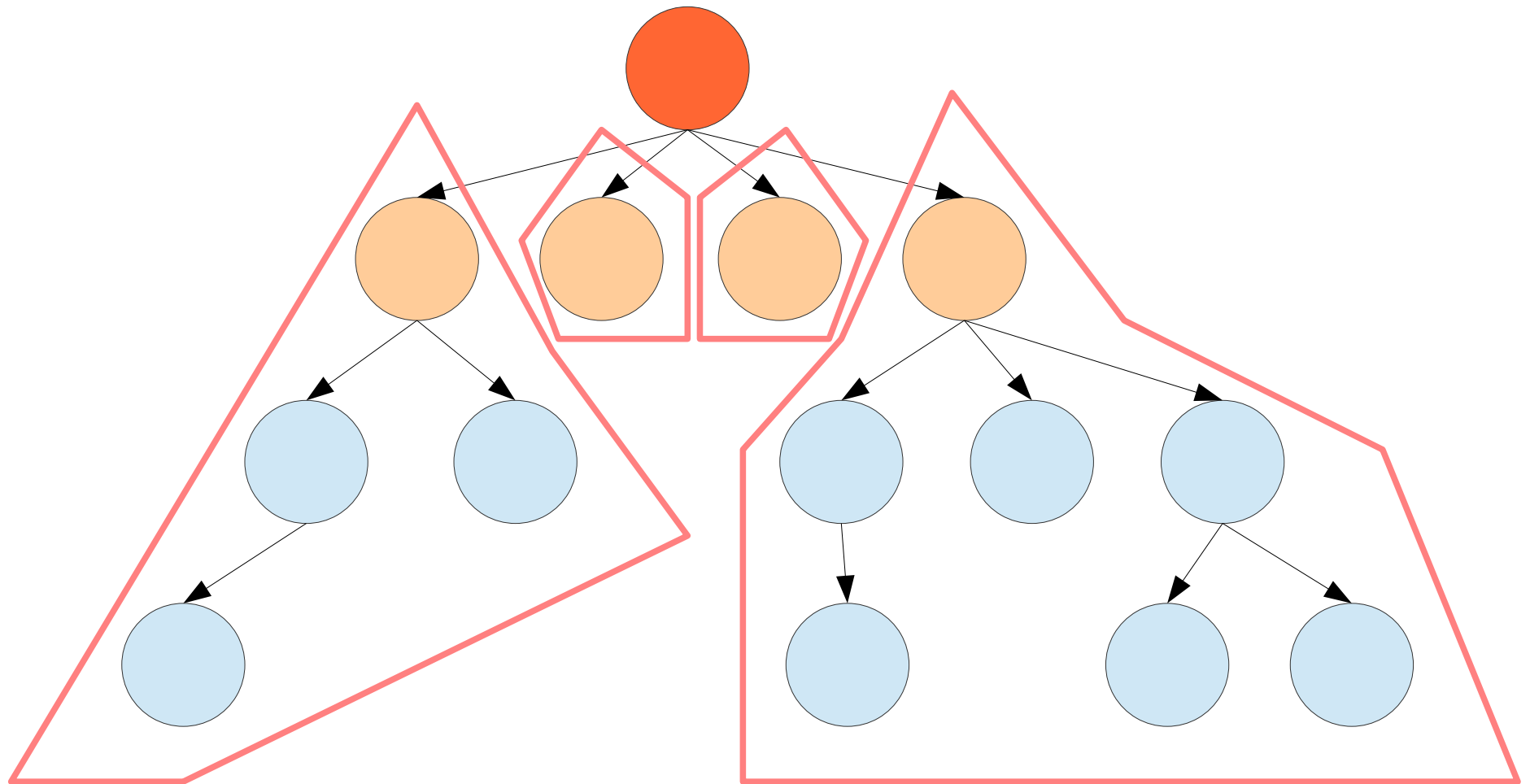


**Ramas o aristas**

# Árboles



# Árboles

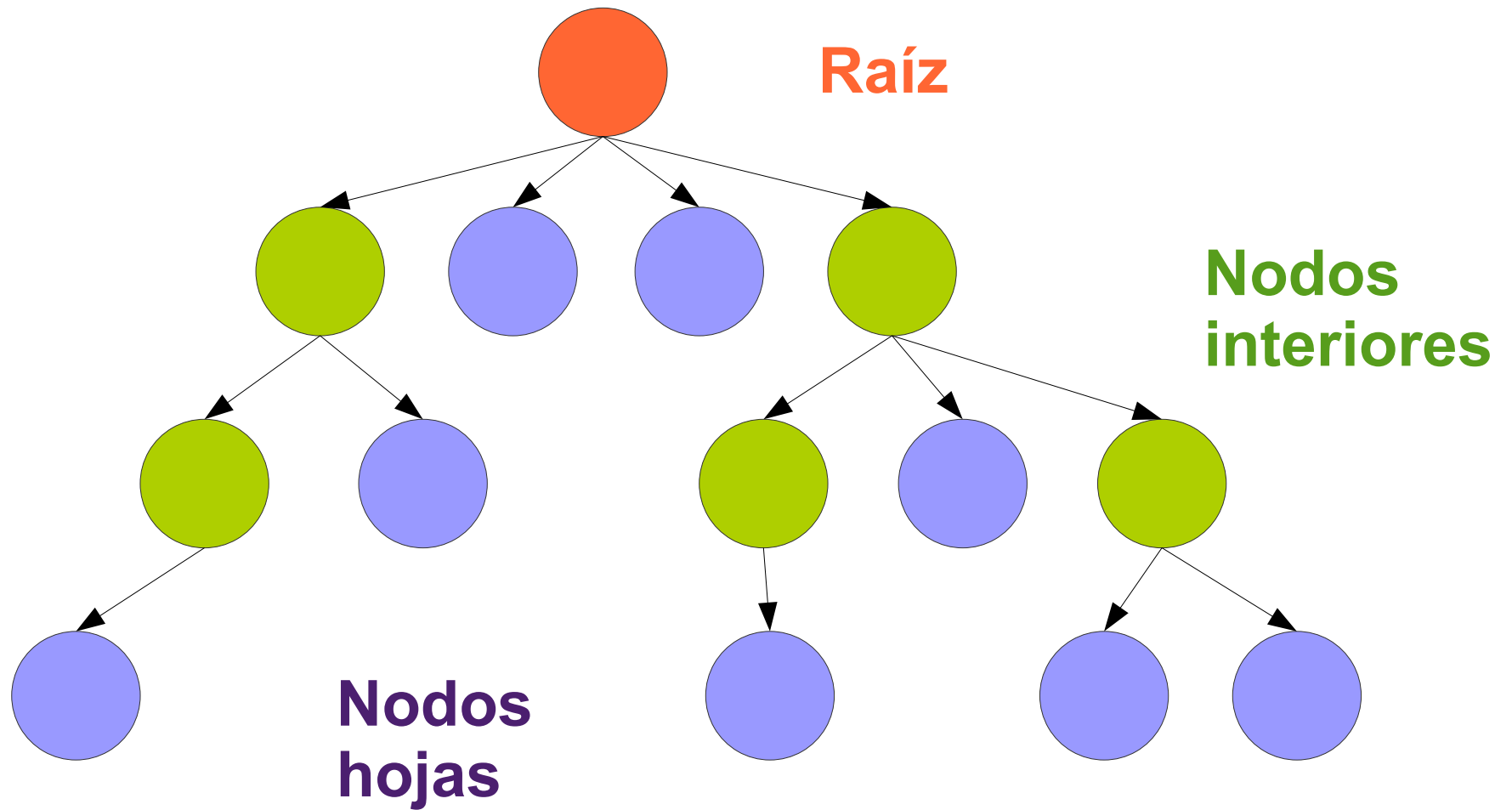


**Subárboles**

# Árboles - terminología

- Nodos: elementos en el árbol.
- Ramas o aristas: conexiones entre los nodos.
- Raíz: nodo especial que es el origen del árbol.
  - Solo existe un nodo raíz en un árbol.
- Nodo hoja: nodo sin una arista o conexión hacia otro nodo.
- Nodo interior: nodo que no es ni raíz ni hoja.

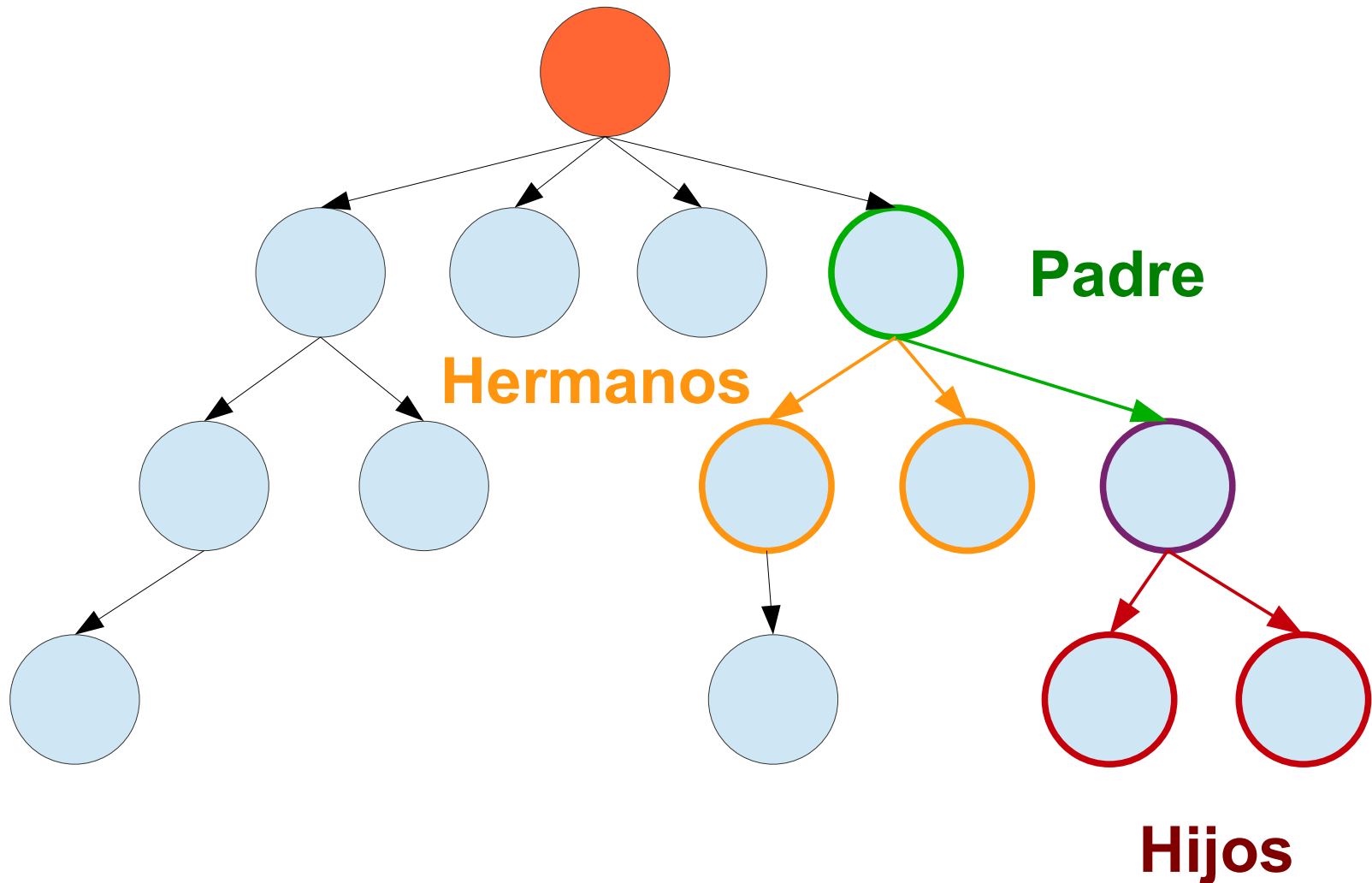
# Árboles - terminología



# Árboles - terminología

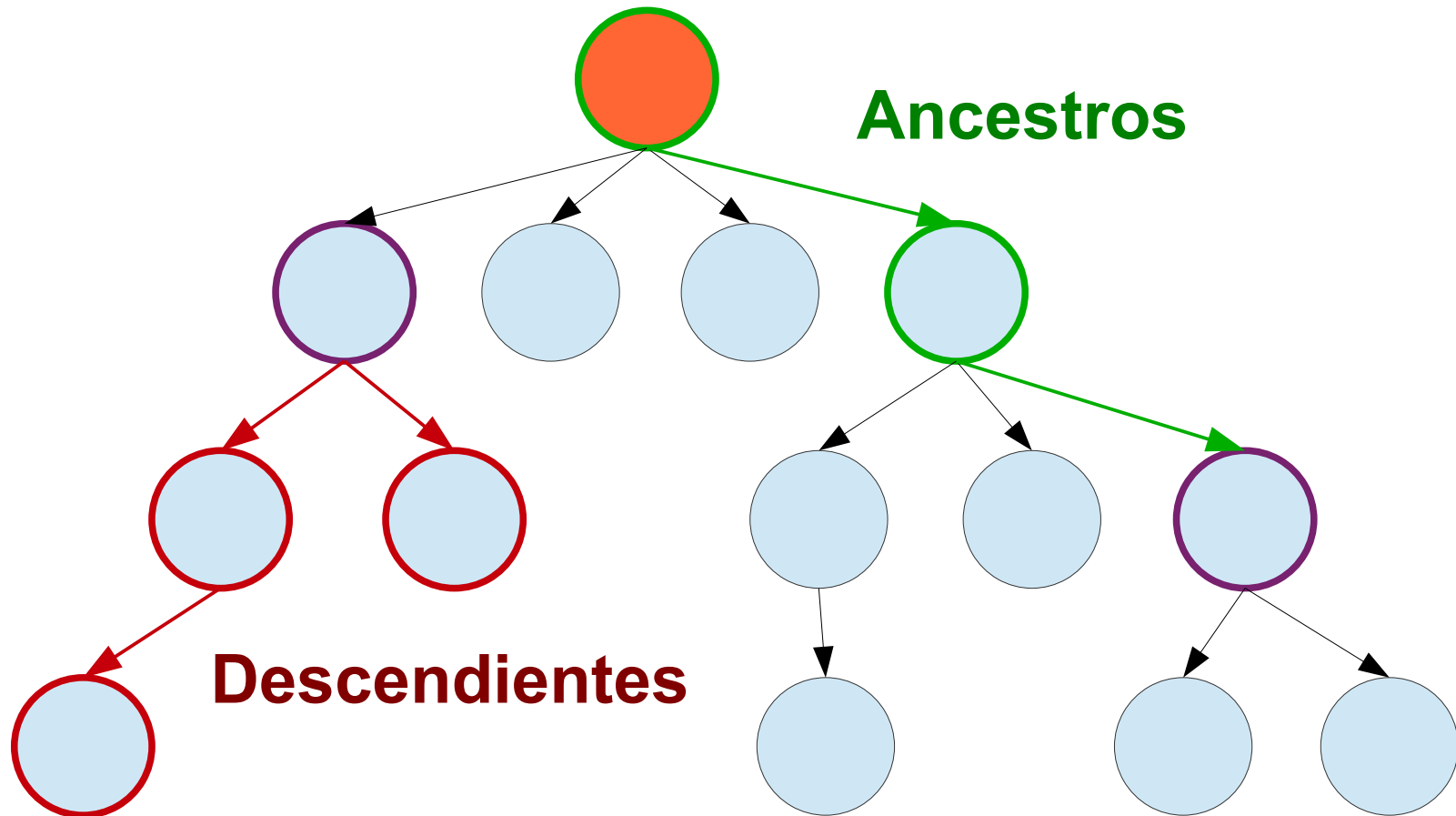
- Nodo padre o predecesor: el nodo directamente encima en la jerarquía.
  - Cada nodo solo tiene un padre o predecesor.
- Nodo hijo o sucesor: el nodo directamente debajo en la jerarquía.
- Nodos hermanos: los que comparten el mismo padre.
- Ancestros de un nodo: el padre, el abuelo, ...
- Descendientes de un nodo: los hijos, los nietos (hijos de los hijos), ...

# Árboles - terminología





# Árboles - terminología



# Árboles - terminología

- Nodos hoja: no tienen ningún hijo.
- Nodo raíz: no tiene padre.
- Nodos internos: tienen un único padre y al menos un hijo.

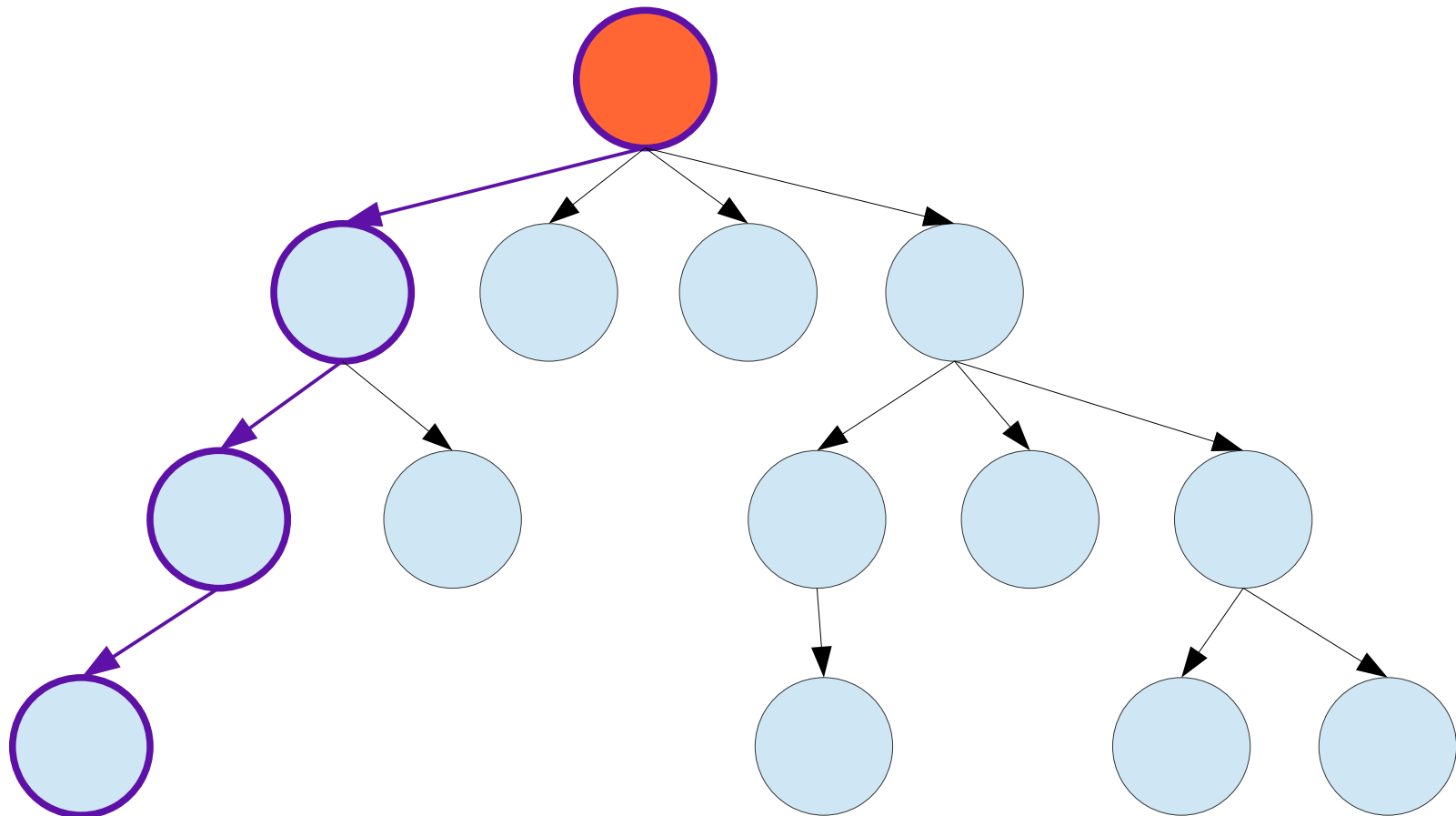
# Árboles - terminología

- Camino: secuencia de aristas o conexiones que llevan de un nodo a otro.
- Longitud de un camino: número de aristas o conexiones en el camino.
- Altura o profundidad de un árbol (no vacío): longitud del camino más largo desde la raíz hacia un nodo hoja.
  - Altura de un árbol vacío: por convención es -1.
  - Altura de un árbol con solo raíz: 0.

# Árboles - terminología

- Altura: puede definirse de forma recurrente:
  - Altura de un nodo hoja: 0.
  - Altura de cualquier otro nodo:  
 $1 + \text{altura máxima de sus subárboles.}$

# Árboles - terminología



**Camino de longitud 3**  
→ **árbol de altura 3**

# Árboles - terminología

- Nivel de un nodo: distancia (longitud del camino) entre el nodo y la raíz.

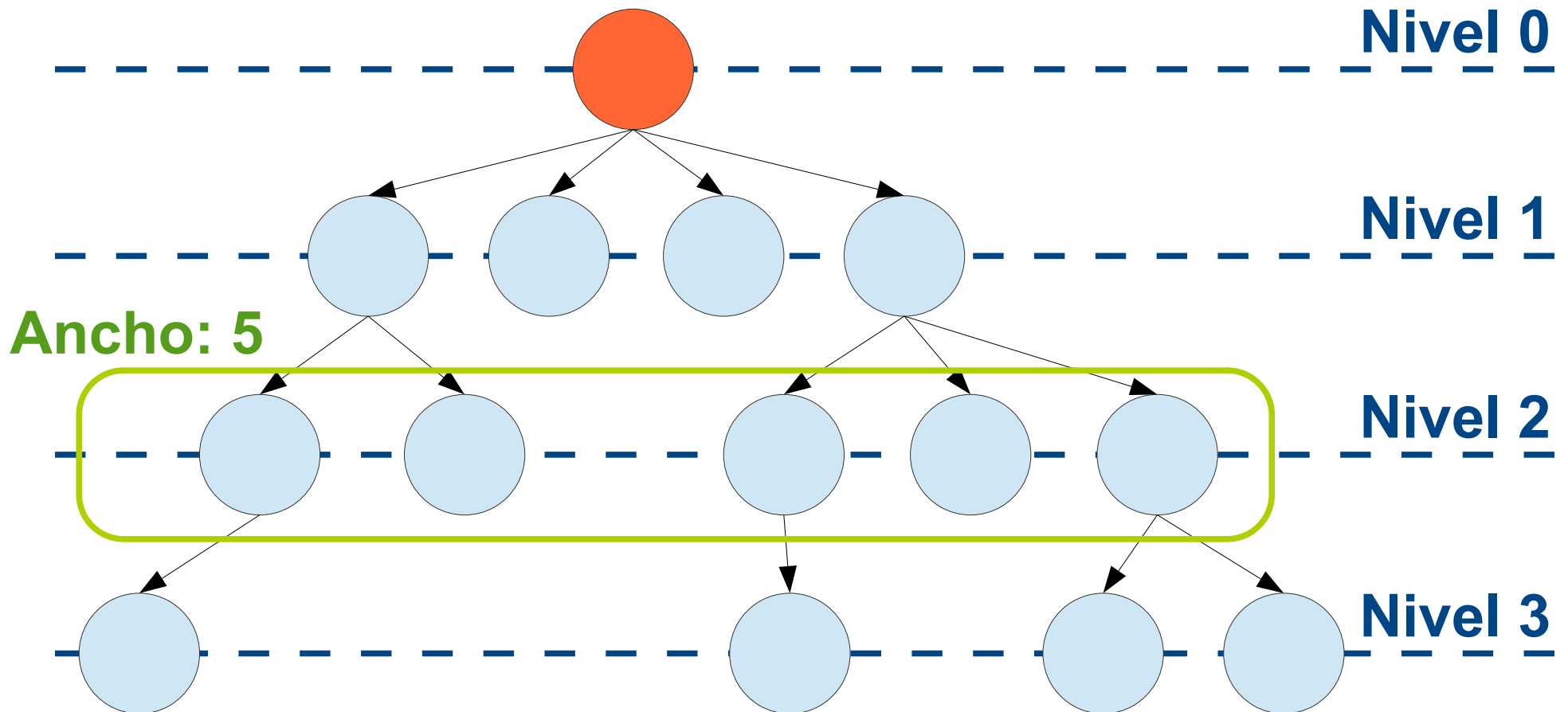
Puede definirse de forma recurrente:

- Nivel del nodo raíz es 0.
  - Nivel de cualquier otro nodo es el **nivel de su padre + 1**.
- Altura del árbol puede definirse como el nivel máximo de los nodos.

# Árboles - terminología

- Nodos hermanos siempre están al mismo nivel, pero en un mismo nivel no todos los nodos son hermanos.
- Ancho de un árbol: número de nodos en el nivel con más nodos.

# Árboles - terminología

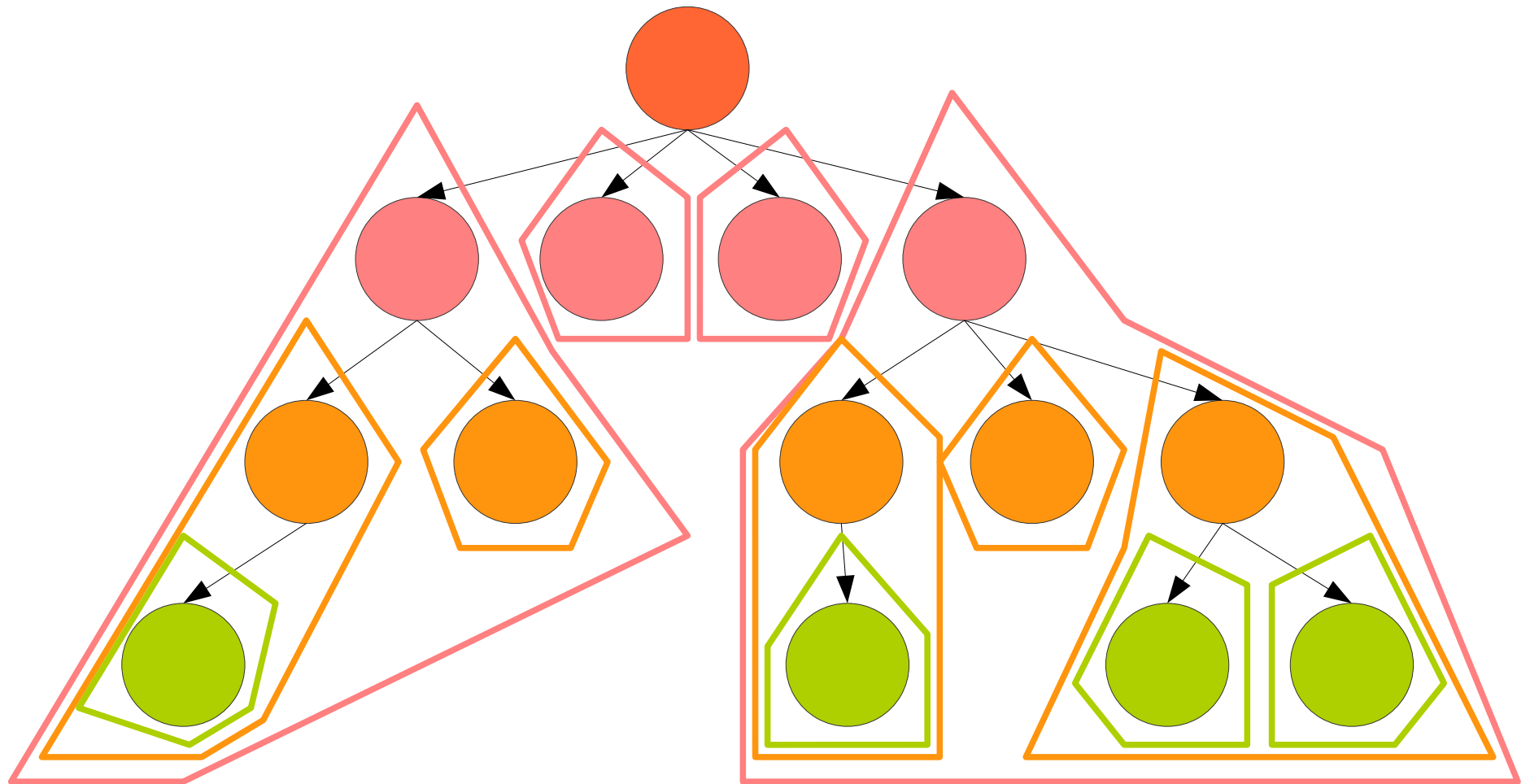




# Árboles - terminología

- Subárbol: cualquier estructura conectada por debajo de la raíz.
- Cada nodo del árbol es la raíz de un subárbol  
→ nodo + todos sus descendientes.
- Definición recurrente de árbol:  
Es un conjunto de nodos que:
  - O bien es vacío.
  - O bien tiene un nodo raíz del que jerárquicamente descienden cero o más subárboles, que son también árboles.

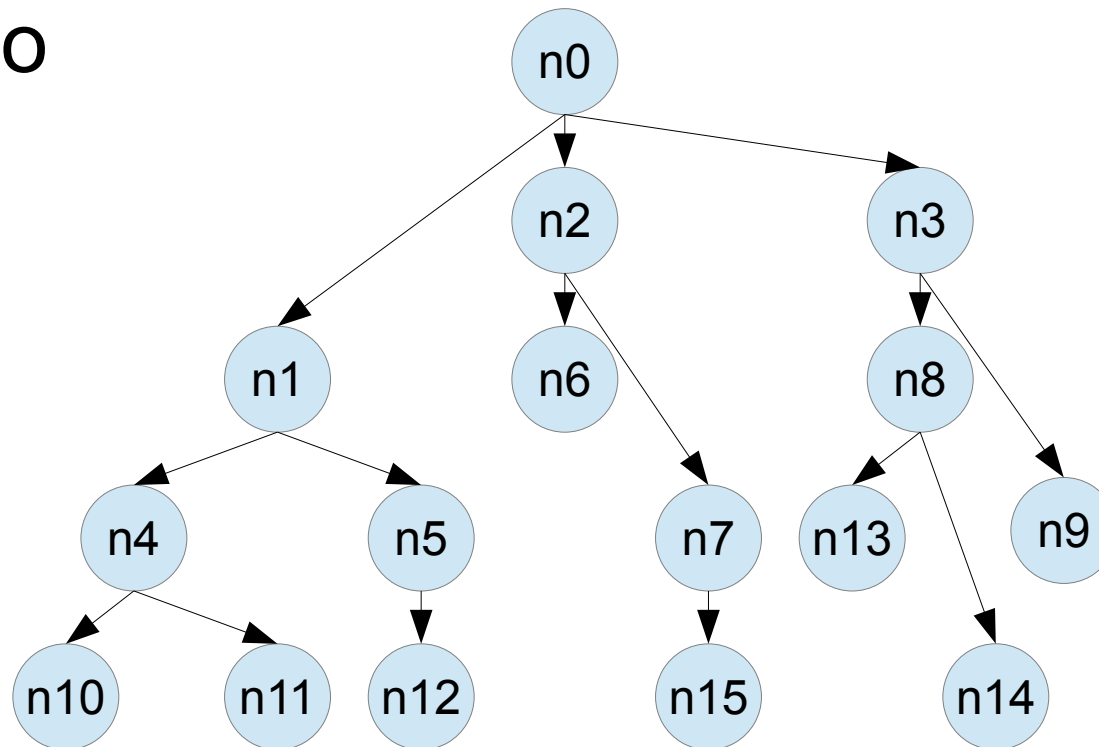
# Árboles - terminología



**Subárboles**

# Árboles - terminología

- Ejercicio

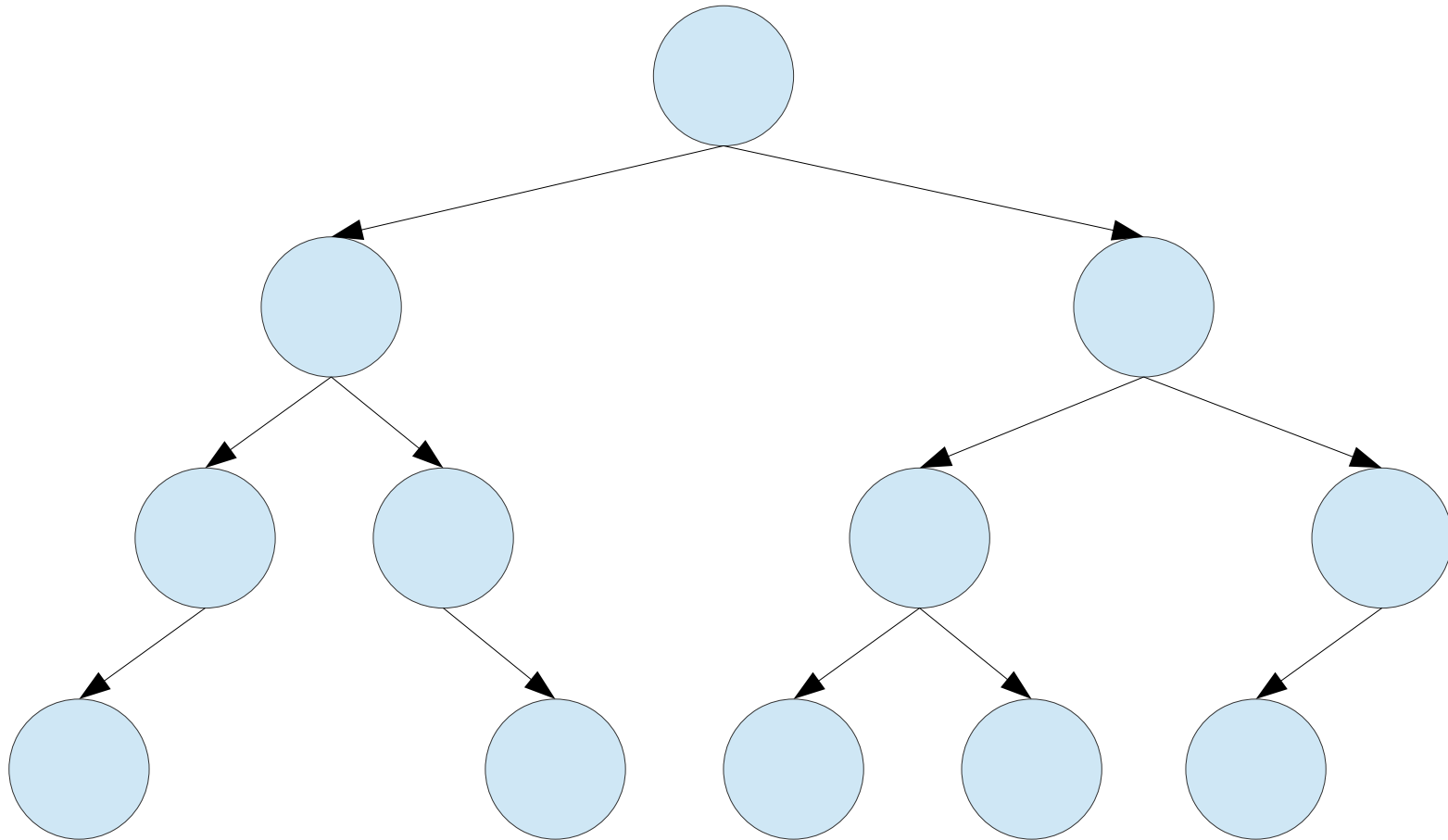


- ¿Nodos de nivel 2?
- ¿Hermanos de n8?
- ¿Subárbol en n3?
- ¿Nodos hojas?
- ¿Hijos de n2?
- ¿Altura del árbol?

# Árboles - terminología

- Grado (orden) de un nodo: el número de hijos que tiene.
- Grado (orden) de un árbol: el máximo de los grados de los nodos del árbol.
- Árbol equilibrado: cuando con un grado de árbol  $k$ , y siendo  $h$  la altura del árbol, cada nodo de nivel  $l < h-1$  tiene exactamente grado  $k$ .

# Árboles - terminología

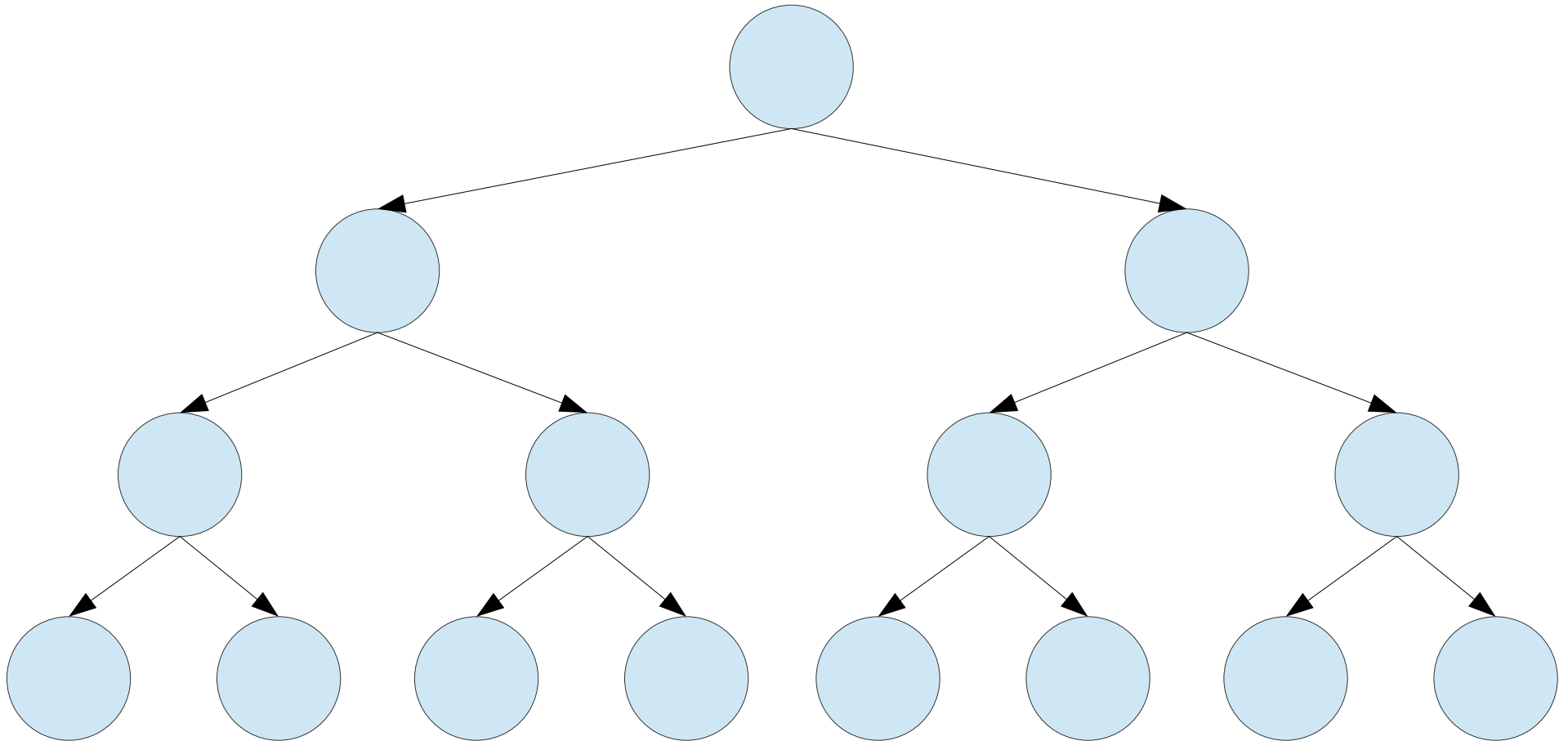


**$k = 2 \rightarrow$  árbol equilibrado**

# Árboles - terminología

- Grado (orden) de un nodo: el número de hijos que tiene.
- Grado (orden) de un árbol: el máximo de los grados de los nodos del árbol.
- Árbol perfectamente equilibrado: cada nodo de nivel  $l < h$  tiene exactamente grado  $k$ .

# Árboles - terminología



**$k = 2 \rightarrow$  árbol perfectamente equilibrado**

# Árboles - terminología

- Orden: propiedad de ordenamiento de los hijos de un nodo.
- Orden local: característica de ordenamiento de los hijos de un nodo dependiendo de su ubicación en el árbol.
- Peso: valor particular que se le asigna al nodo y/o a los subárboles.



# Recorridos en árboles

- Recorrido: avanzar a través de los nodos del árbol, utilizando las ramas o aristas.

Visitar una sola vez todos los nodos del árbol, siguiendo algún orden particular.

Los nodos pueden atravesarse varias veces, pero solo se visitan una vez.

# Recorridos en árboles

- Recorridos en profundidad: seleccionar un camino y visitar todos los nodos hasta un nodo hoja, antes de cambiar de camino.
  - Recorrido en pre-orden: visitar el nodo padre antes que todos los nodos hijos.
  - Recorrido en pos-orden: visitar todos los nodos hijos antes que su respectivo nodo padre.

# Recorridos en árboles

- Recorrido en in-orden (principalmente para árboles binarios): visitar primero el subárbol izquierdo, luego el nodo padre, y finalmente el subárbol derecho.

# Recorridos en árboles

- Recorrido en anchura: visitar todos los hermanos a la misma altura antes de avanzar a un siguiente nivel.
  - Recorrido por niveles: visitar el nodo padre, luego todos sus hijos directos, luego todos sus nietos, ...

# Tipos de árboles

- Árbol general: árbol en el cual cada uno de sus nodos puede tener cualquier número de hijos.
- Árbol  $n$ -ario ( $n$ -árbol): árbol en el cual cada uno de sus nodos puede tener no más de  $n$  hijos (árbol con grado  $n$ ).
- Árbol  $n$ -ario ( $n$ -árbol) completo: árbol en el cual todos los nodos distintos de las hojas tienen exactamente  $n$  hijos.

# Diseño e Implementación

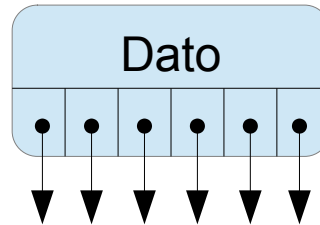
## TAD ArbolGeneral

# Representación de árboles

- Elemento base de construcción: Nodo.
- Nodo contiene:
  - Un dato (primario o definido por el usuario).
  - Una lista (arreglo, ...) de vínculos (apuntadores) a los Nodos hijos, ordenados de izquierda a derecha.
- Un árbol general se define entonces como un apuntador al Nodo de inicio (raíz).

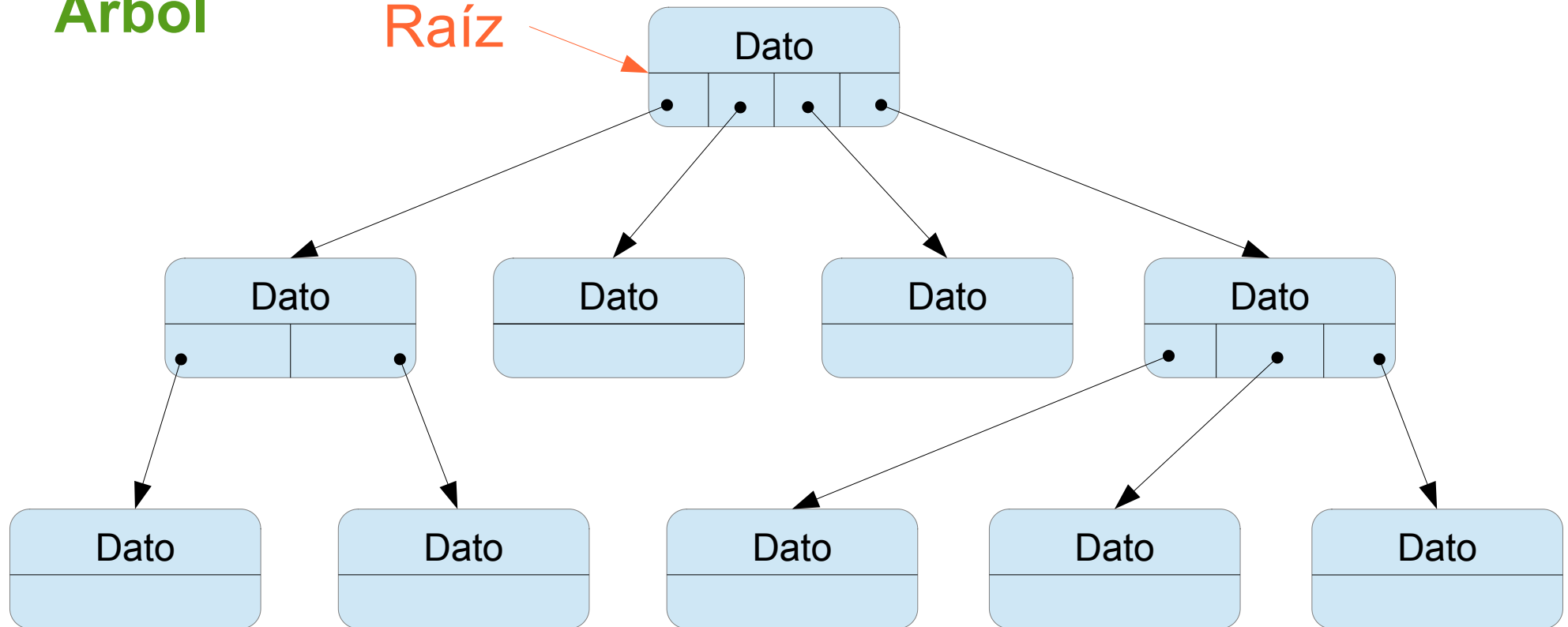
# Representación de árboles

Nodo



Árbol

Raíz





# TAD NodoGeneral

- Auxiliar para la definición de un árbol.
- Conjunto mínimo de datos:
  - dato del nodo.
  - lista de apuntadores a nodo (para los hijos).
- Operaciones:
  - constructores, destructores.
  - asignar, extraer el dato del nodo.
  - asignar, extraer una referencia a nodo (hijo).

# NodoGeneral.h

```
template< class T >
class NodoGeneral {
    protected:
        T dato;
        std::list< NodoGeneral<T>* > desc;

    public:
        NodoGeneral();
        ~NodoGeneral();
        T& obtenerDato();
        void fijarDato(T& val);
        void limpiarLista();
        void adicionarDesc(T& nval);
        void eliminarDesc(T& val);
};
```

# TAD ArbolGeneral

- Estado: conjunto mínimo de datos.
  - Nodo raíz.
- Interfaz: comportamiento (operaciones).
  - constructores, destructores.
  - asignación, extracción del nodo raíz.
  - verificación de árbol vacío.
  - tamaño del árbol (en nodos), altura del árbol.
  - inserción, eliminación de un nodo.
  - verificación de un nodo en el árbol.
  - recorridos sobre el árbol.

# ArbolGeneral.h

```
template <class T>
class ArbolGeneral {
    protected:
        NodoGeneral<T>* raiz;

    public:
        ArbolGeneral();
        ArbolGeneral(T& val);
        ~ArbolGeneral();
        bool esVacio();
        NodoGeneral<T>* obtenerRaiz();
        void fijarRaiz(NodoGeneral<T>* nraiz);
        bool insertarNodo(T& padre, T& n);
        bool eliminarNodo(T& n);
```

# ArbolGeneral.h

```
bool buscar(T& n);  
unsigned int altura(...);  
unsigned int tamaño(...);  
void preOrden(...);  
void posOrden(...);  
void nivelOrden(...);  
};
```

# Tarea

- Revisar los videos de la siguiente lista de reproducción:  
<https://youtube.com/playlist?list=PLXNX3bqAk3JNXGbHsEF-NPTjXg67qOZk9>
- Realizar la implementación propia (por grupos) del TAD ArbolGeneral, siguiendo los lineamientos explicados en los videos.

# Referencias

- [en.wikipedia.org/wiki/Tree\\_\(data\\_structure\)](https://en.wikipedia.org/wiki/Tree_(data_structure))
- [www.cs.nmsu.edu/~epontell/courses/cs272/ disp/trees/2004/tree2004.pdf](http://www.cs.nmsu.edu/~epontell/courses/cs272 disp/trees/2004/tree2004.pdf)
- [www.csd.uwo.ca/~vmazalov/CS1027a/notes/CS1027-Trees\\_6up.pdf](http://www.csd.uwo.ca/~vmazalov/CS1027a/notes/CS1027-Trees_6up.pdf)
- [people.cis.ksu.edu/~schmidt/300s05/ Lectures/Week7b.html](http://people.cis.ksu.edu/~schmidt/300s05/Lectures/Week7b.html)
- [pages.cs.wisc.edu/~vernon/cs367/notes/8.TREES.html](http://pages.cs.wisc.edu/~vernon/cs367/notes/8.TREES.html)