

01 – PROGRAMACIÓN I

MG. NICOLÁS BATTAGLIA



UAIOnline
ultra >>>

UNIDAD 3

ESTRUCTURAS DINÁMICAS 2 - CLASE 8 - ARBOLES



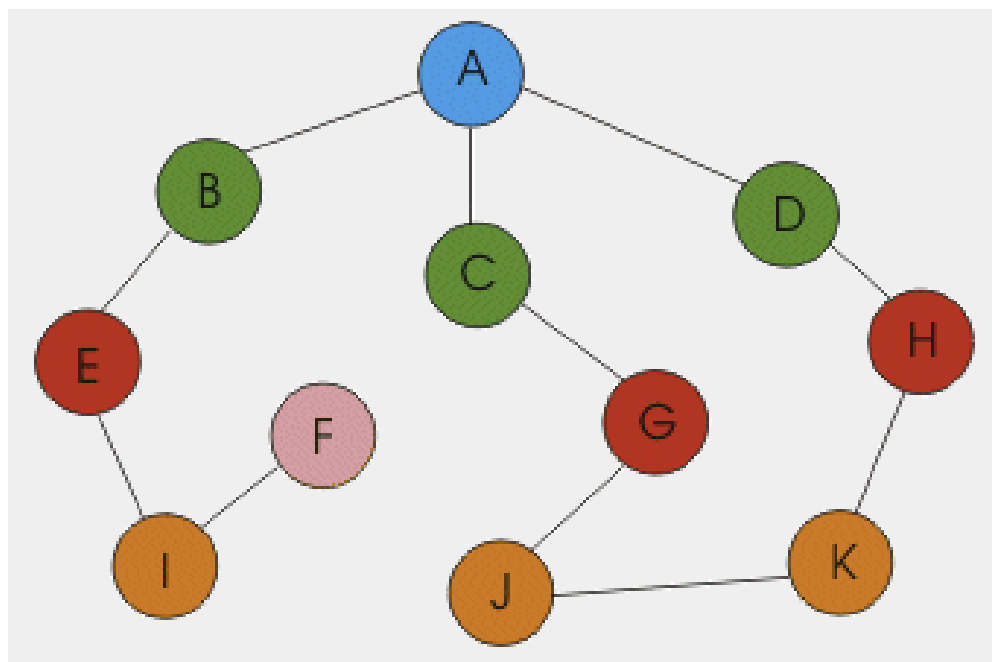
UAIOnline
ultra >>>

ARBOLES

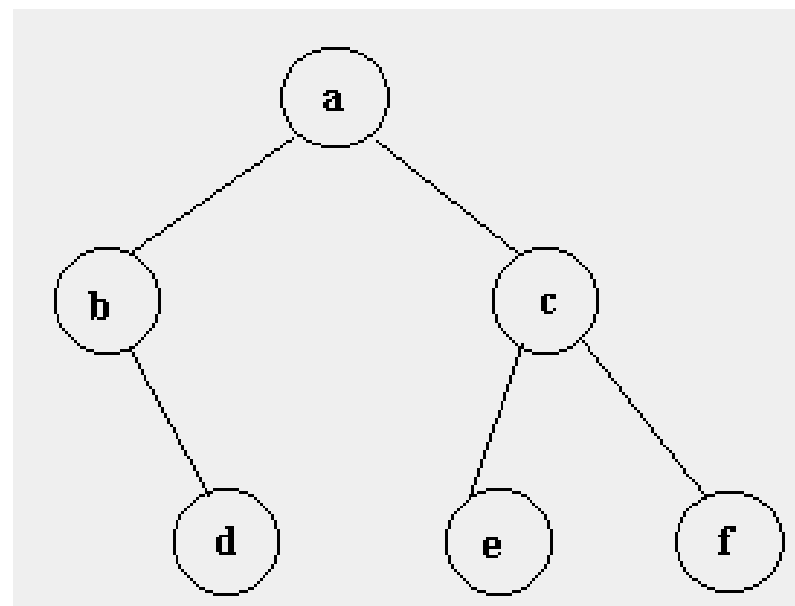
- Un árbol es una estructura de datos, que puede definirse de forma recursiva como
 - Una estructura vacía
 - Un elemento o clave de información (nodo) más un número finito de estructuras tipo árbol, disjuntos, llamados subárboles.
- Si dicho número de estructuras es inferior o igual a 2, se tiene un árbol binario.
- Un árbol es, por tanto, una estructura no secuencial.
- Otra definición nos da el árbol como un tipo de grafo acíclico, conexo y no dirigido.
 - Es un grafo no dirigido en el que existe exactamente un camino entre todo par de nodos.
 - Esta definición permite implementar un árbol y sus operaciones empleando las representaciones que se utilizan para los grafos.

ARBOLES

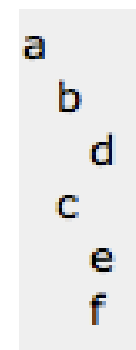
Grafo



Árbol representado como grafo

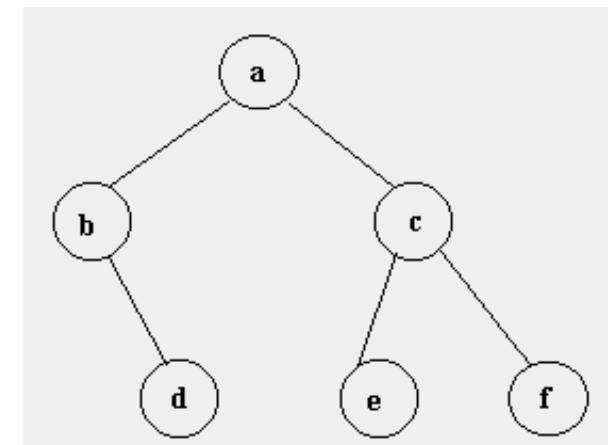


**Árbol
representado
como un
diagrama en
columnas**



ARBOLES AVL

- Árbol binario balanceado (Árbol AVL – Por sus siglas Adelson, Velski y Landis, sus creadores)
- Un árbol balanceado simplifica en gran medida las operaciones de búsqueda
 - Está balanceado cuando para todos sus nodos la altura de la rama izquierda no difiere más de una unidad de la altura de la rama derecha (o viceversa)
 - El factor de equilibrio es la diferencia entre las alturas de un subárbol con las del otro subárbol: $FE=0, -1$ o 1 .
 - $FE=0$ -> Nodo perfectamente equilibrado
 - $FE=1$ -> Nodo derecho es un nivel más alto
 - $FE=-1$ -> Nodo izquierdo es un nivel más alto
 - $FE \geq |2|$ -> Hay que equilibrar el árbol
- Para equilibrar o balancear un árbol hay que aplicar rotaciones

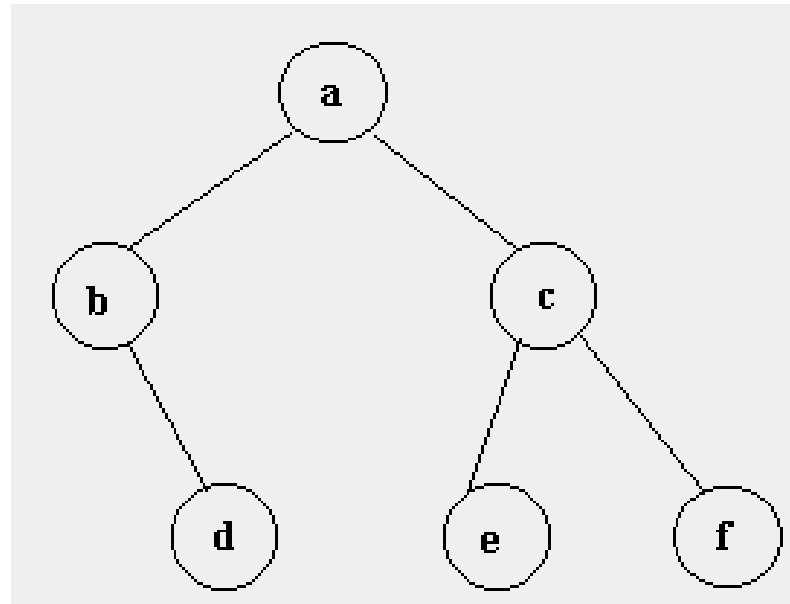


ARBOLES – NOMENCLATURA

- **Raíz:** es aquel elemento que no tiene antecesor; ejemplo: a .
- **Rama:** arista entre dos nodos.
- **Antecesor:** un nodo X es antecesor de un nodo Y si por alguna de las ramas de X se puede llegar a Y .
- **Sucesor:** un nodo X es sucesor de un nodo Y si por alguna de las ramas de Y se puede llegar a X .
- **Grado de un nodo:** el número de descendientes directos que tiene.
- **Hoja:** nodo que no tiene descendientes: grado 0.
- **Nodo interno:** aquel que tiene al menos un descendiente.
- **Nivel:** número de ramas que hay que recorrer para llegar de la raíz a un nodo.
- **Altura:** el nivel más alto del árbol.
- **Anchura:** es el mayor valor del número de nodos que hay en un nivel.

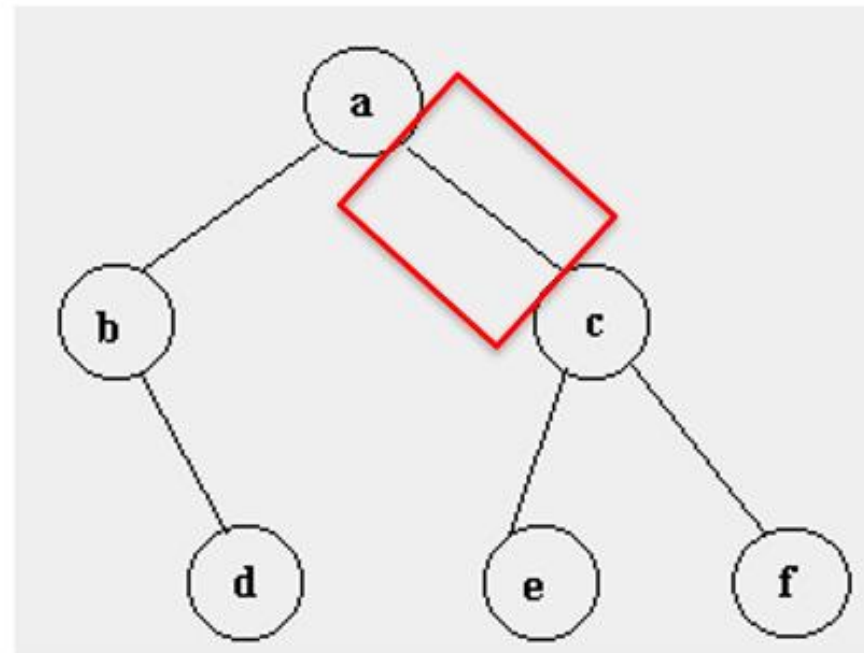
ARBOLES - NOMENCLATURA

- **Raíz:** es aquel elemento que no tiene antecesor; ejemplo: *a*.



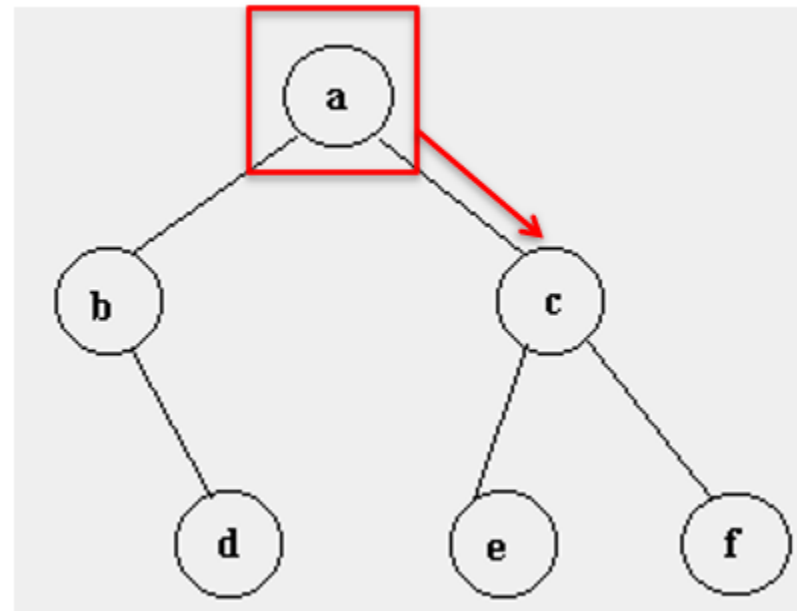
ARBOLES - NOMENCLATURA

- **Rama:** arista entre dos nodos.



ARBOLES - NOMENCLATURA

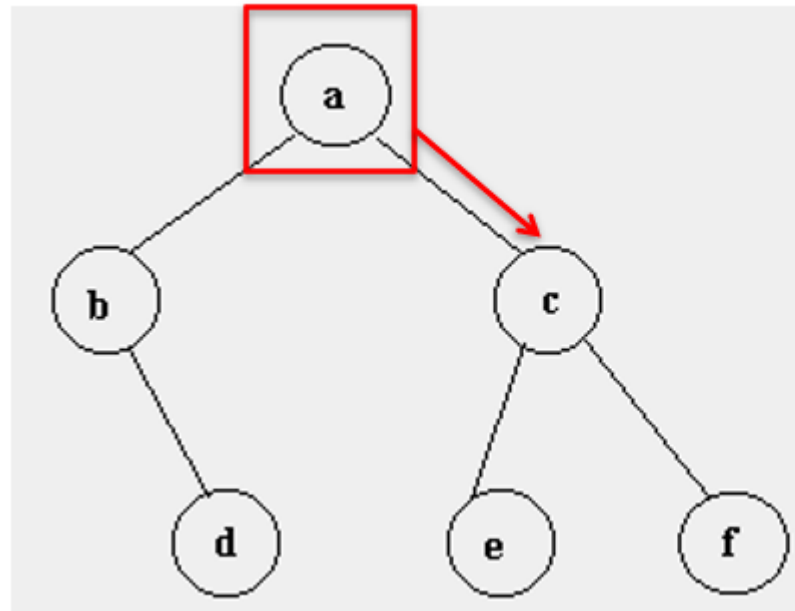
- **Antecesor:** un nodo X es antecesor de un nodo Y si por alguna de las ramas de X se puede llegar a Y.



“a” es antecesor
de “c” pues se
puede llegar a
“c” desde “a” por
una arista

ARBOLES - NOMENCLATURA

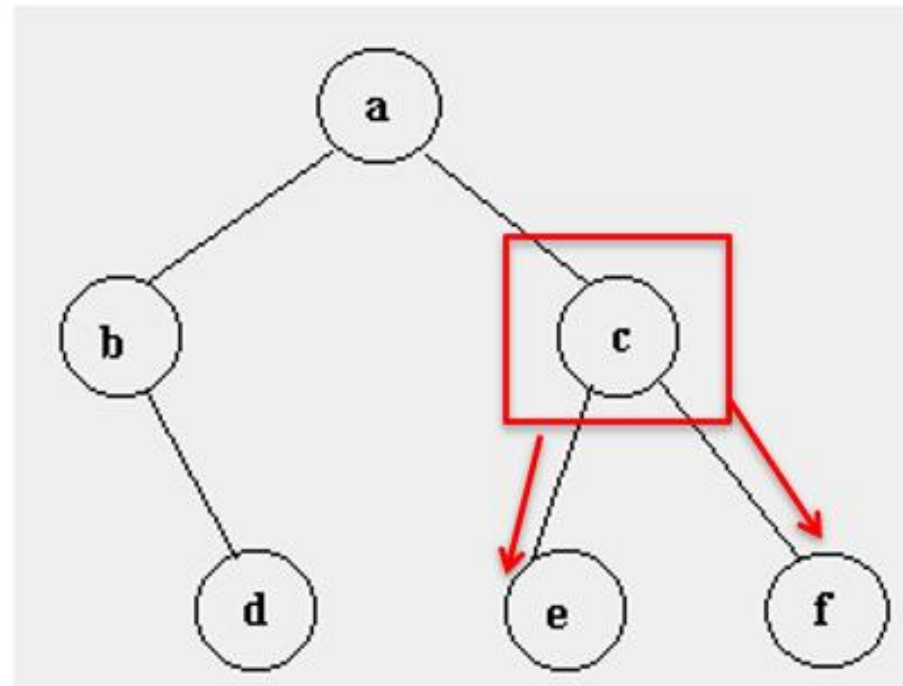
- **Sucesor:** un nodo X es sucesor de un nodo Y si por alguna de las ramas de Y se puede llegar a X.



“c” es sucesor de
“a” pues se
puede llegar a
“c” desde “a” por
una arista

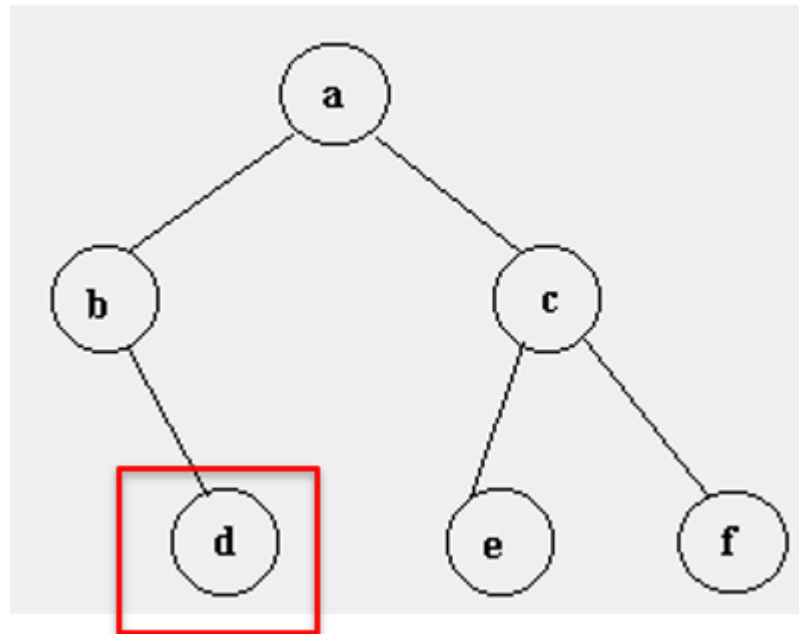
ARBOLES - NOMENCLATURA

- **Grado de un nodo:** el número de descendientes directos que tiene. Ejemplo: *c* tiene grado 2, *d* tiene grado 0, *a* tiene grado 2.



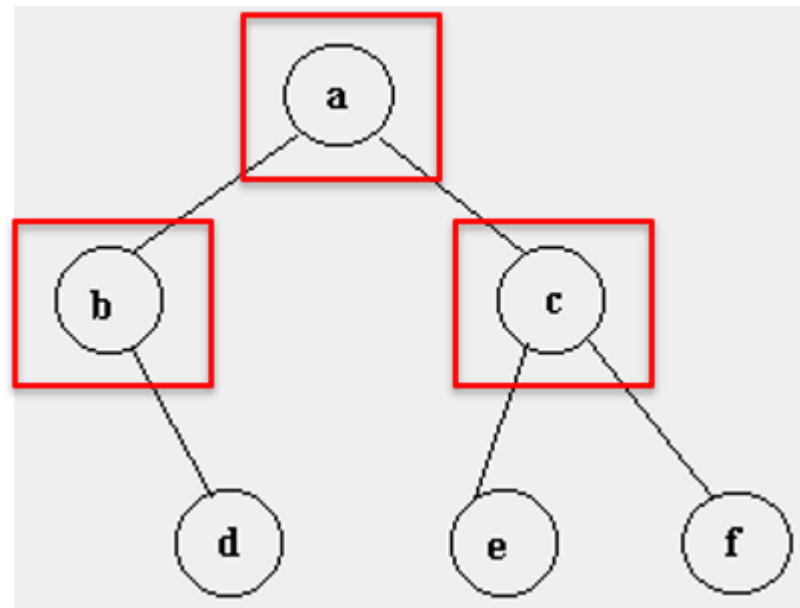
ARBOLES - NOMENCLATURA

- **Hoja:** nodo que no tiene descendientes: grado 0. Ejemplo: *d*



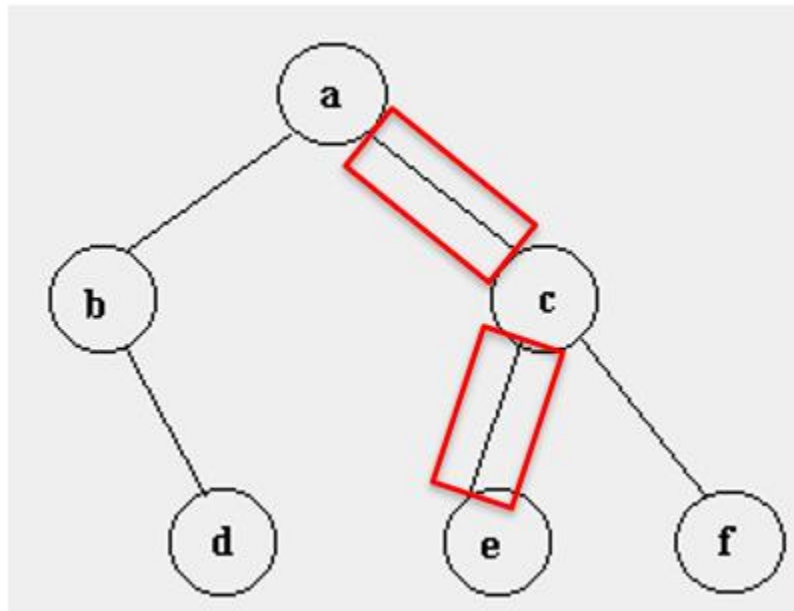
ARBOLES - NOMENCLATURA

- **Nodo interno:** aquel que tiene al menos un descendiente.



ARBOLES - NOMENCLATURA

- **Nivel:** número de ramas que hay que recorrer para llegar de la raíz a un nodo. Ejemplo: el nivel del nodo *a* es 1 (es un convenio), el nivel del nodo *e* es 3.



ARBOLES - NOMENCLATURA

- **Anchura:** es el mayor valor del número de nodos que hay en un nivel. En la figura, la anchura es 3.

