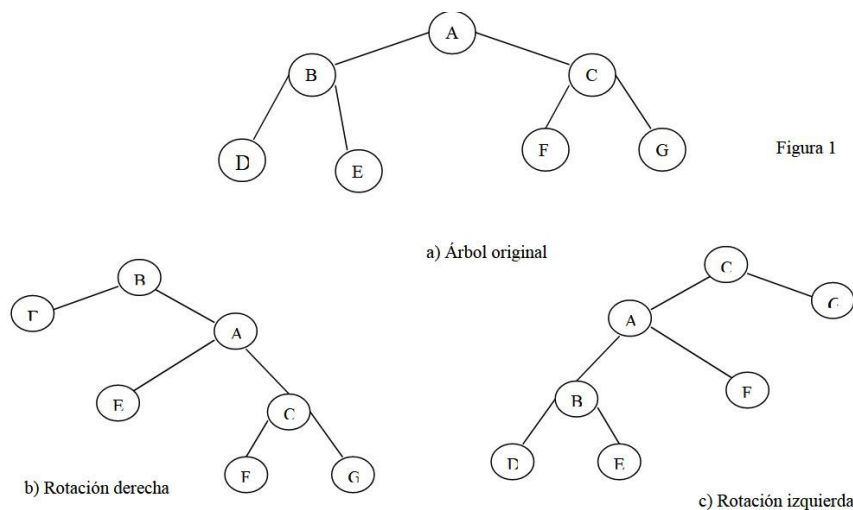


## Árbol balanceado

La búsqueda más eficiente se efectúa en un árbol binario balanceado.

Desafortunadamente, la función Inserta no asegura que el árbol permanezca balanceado, el grado de balance depende del orden en que son insertados los nodos en el árbol. La altura de un árbol binario es el nivel máximo de sus hojas (profundidad). La altura del árbol nulo se define como  $-1$ . Un árbol binario balanceado es un árbol binario en el cual las alturas de los dos subárboles de todo nodo difiere a lo sumo en 1. El balance de un nodo en un árbol binario se define como la altura de su subárbol izquierdo menos la altura de su subárbol derecho. Cada nodo en un árbol binario balanceado tiene balance igual a 1, -1 o 0, dependiendo de si la altura de su subárbol izquierdo es mayor que, menor que o igual a la altura de su subárbol derecho. Supóngase que tenemos un árbol binario balanceado, y usamos la función para insertar un nodo en dicho árbol. Entonces el árbol resultante puede o no permanecer balanceado. Es fácil ver que el árbol se vuelve desbalanceado si y solo si el nodo recién insertado es un descendiente izquierdo de un nodo que tenía de manera previa balance de 1, o si es un hijo derecho descendiente de un nodo que tenía de manera previa balance  $-1$ . Para que el árbol se mantenga balanceado es necesario realizar una transformación en el mismo de manera que:

1. El recorrido en orden del árbol transformado sea el mismo que para el árbol original (es decir, que el árbol transformado siga siendo un árbol de búsqueda binaria).
2. El árbol transformado esté balanceado.



El árbol de la figura 1b es una rotación derecha del árbol con raíz en A de manera similar, el árbol de la figura 1c se dice que es una rotación izquierda del árbol con raíz en A. Un algoritmo para implantar una rotación izquierda de un subárbol con raíz en p es el siguiente:

```
q= right(p);
hold=left(q);
left(q)=p;
right(p)=hold;
```