

Lenguaje natural:

Se va a recorrer un árbol en búsqueda de una etiqueta pasada por parámetros. Por lo tanto, para recorrer el árbol se va a aplicar el recorrido de primer hijo, hermano derecho. Si este se encuentra, devolvemos su etiqueta, pero si no está, devolvemos nulo.

PreCondiciones:

- El árbol no debe estar vacío.

PostCondiciones:

- Si se encuentra un nodo con la etiqueta buscada, se devuelve ese nodo.
- Si no se encuentra el nodo con la etiqueta buscada, se devuelve nulo.
- El árbol no se modifica.

Pseudocódigo y Orden del tiempo de ejecución:

Buscar(unaEtiqueta):

COM

 unhijo <- PrimerHijo; O(1)

 MIENTRAS unhijo <> nulo hacer O(n)

 unHijoBuscar <- unhijo. Buscar(unaEtiqueta); O(1)

 SI unHijoBuscar <> nulo hacer O(1)

 SI (unHijo.getEtiqueta() == unaEtiqueta) O(1)

 return unhijo O(1)

 SINO

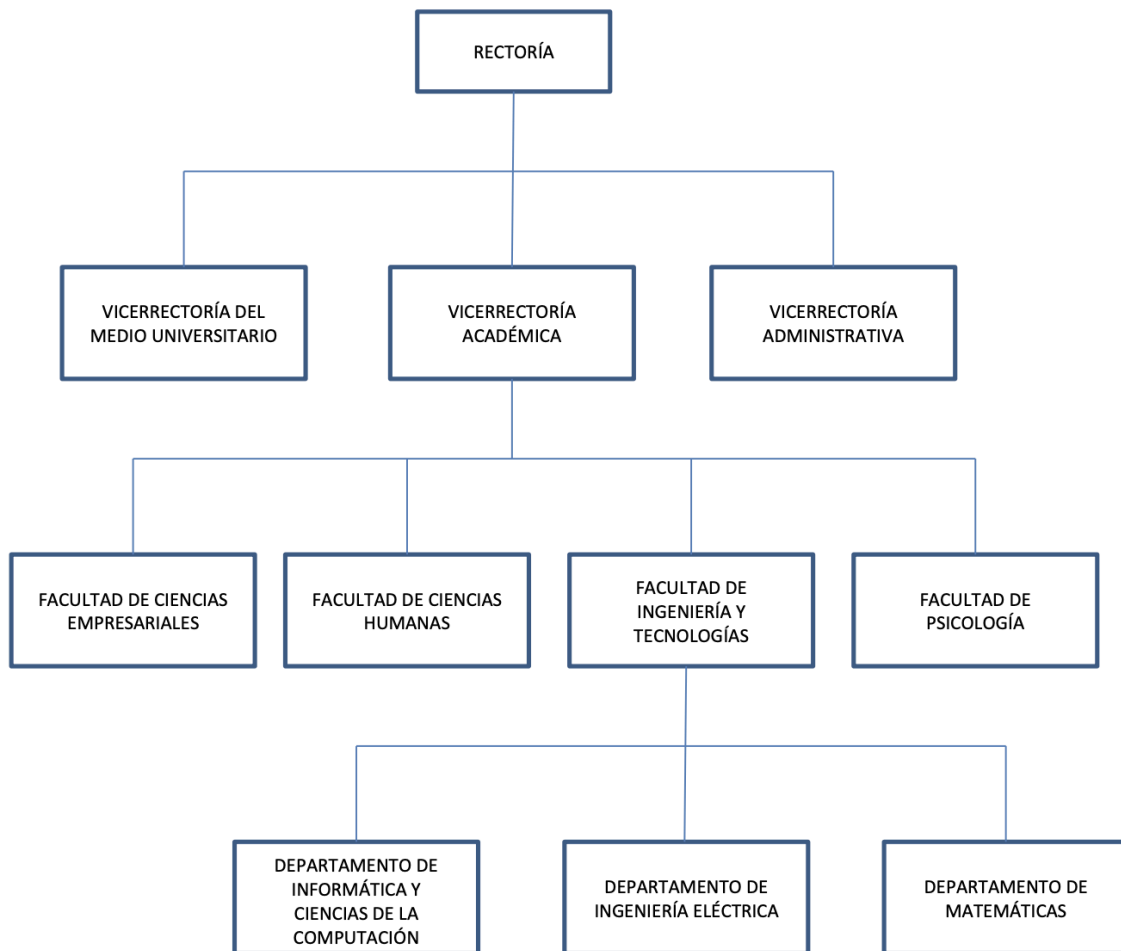
 unHijo <- unHijo.hermanoDerecho; O(1)

 FIN MIENTRAS

 Return nulo

FIN

Orden de ejecución del algoritmo: $O(n)$



Ejemplo de ejecución para encontrar “Facultad de ciencias empresariales”:

- 1 - Rectoría
- 2 - Vicerrectoría del medio universitario
- 3 - Vicerrectoría académica
- 4 - Vicerrectoría administrativa
- 5 - Facultad de ciencias empresariales

Cantidad de comparaciones: 5

Ejemplo para encontrar “Departamento de Medicina”:

Va a recorrer todo el árbol buscando la etiqueta, y va a devolver nulo ya que, no la va a encontrar.

Cantidad de comparaciones para una etiqueta que no existe: n
siendo “n” la cantidad de objetos que tiene el árbol.

g) Orden de tiempo de ejecución para distintos casos:

El orden de tiempo de ejecución para todos los casos va a ser $O(n)$.