

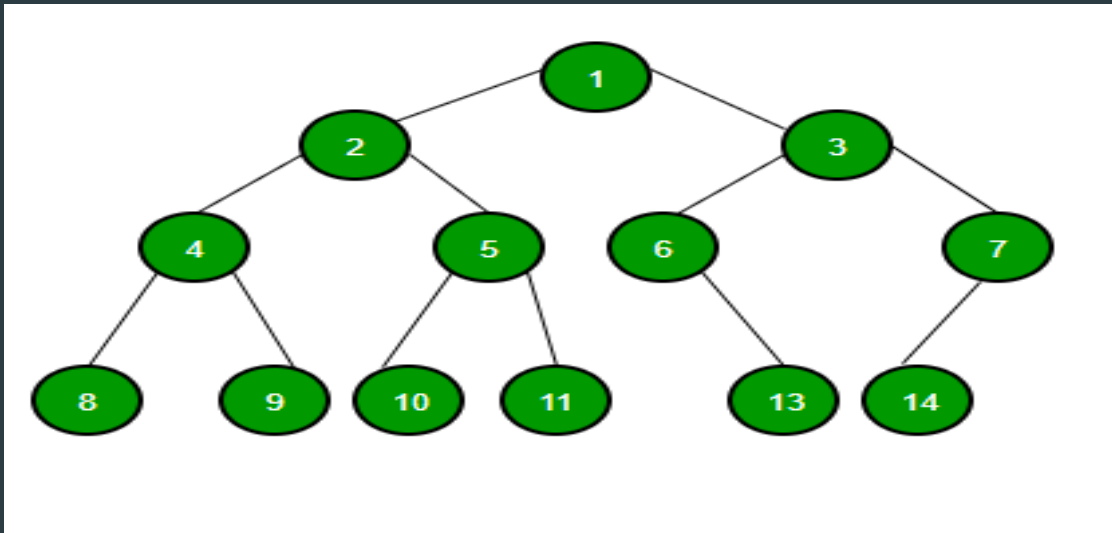
ARBOLES BINARIOS

Monitoria de estructura de datos

Luis F. Muñoz

Que son?

- Un árbol es una estructura de datos dinámica-jerárquica o no lineal, a diferencia de los arreglos, pilas y colas.
- Un árbol que tenga como máximo dos nodos hijos es llamado árbol binario.
- A estos dos hijos los llamaremos Izquierda y Derecha.

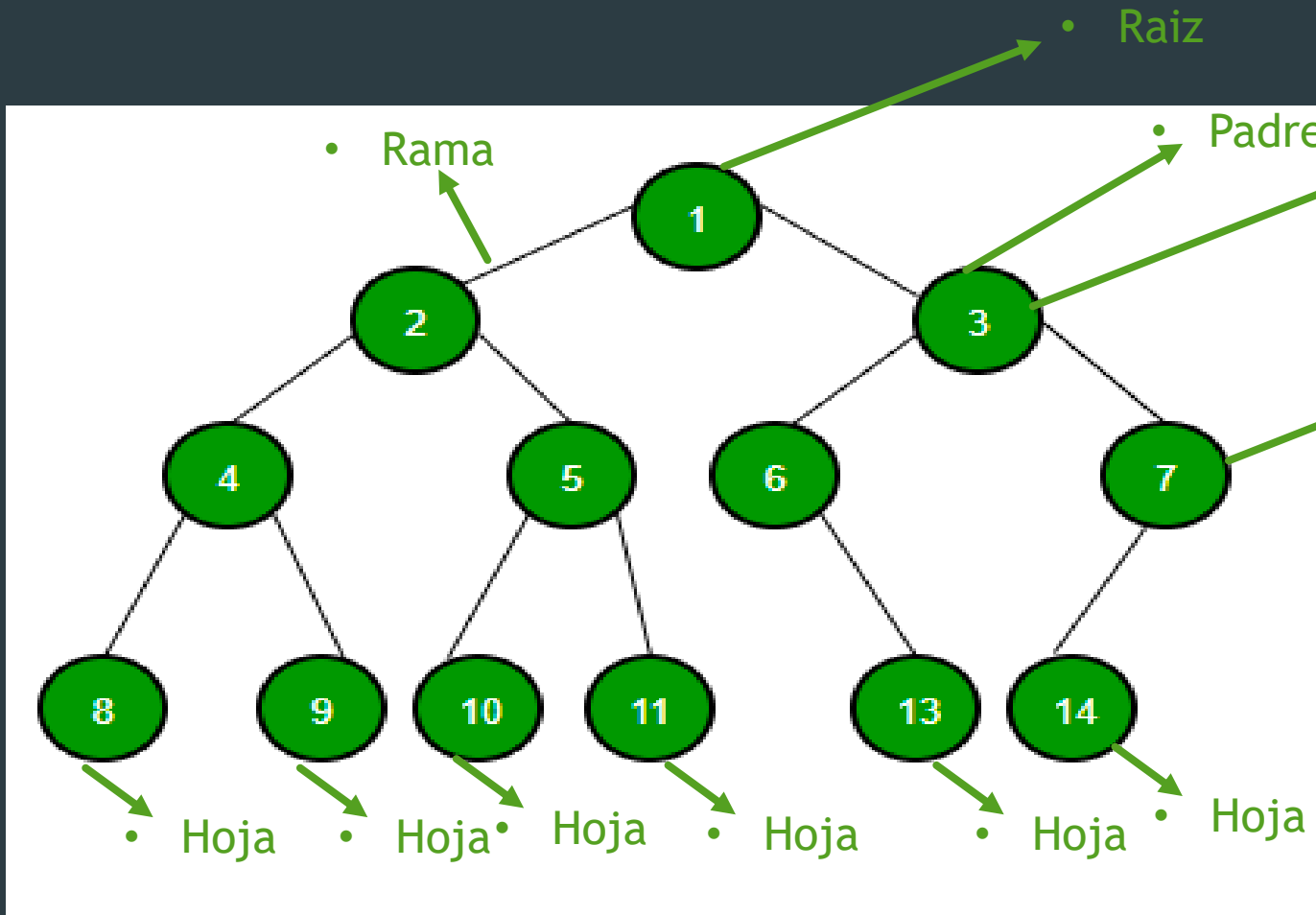


Composición de un árbol binario.

- ▶ Dato
- ▶ Puntero al hijo izquierdo
- ▶ Puntero al hijo derecho

```
Class Nodo {  
    public:  
        Int dato;  
        Nodo *izquierdo;  
        Nodo *derecho;  
}
```

Vocabulario de los arboles binarios.



□ Altura-p: cantidad de niveles.

□ Nivel: distancia desde la raíz.

Por que los arboles?

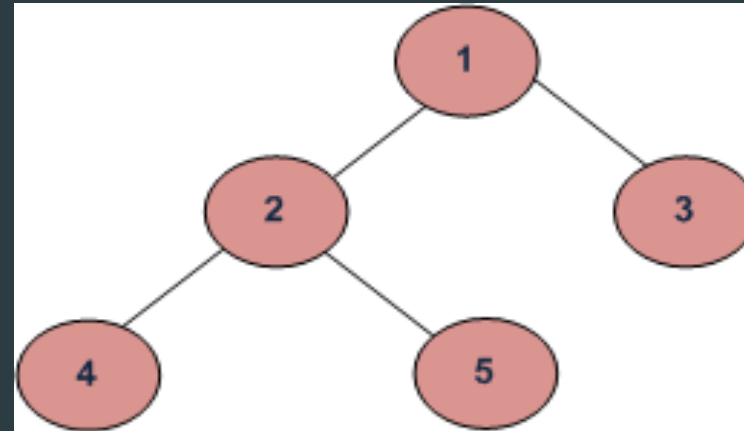
- ▶ Para almacenar información de manera jerárquica.
- ▶ Acceso/Búsqueda (lista enlazada > BST < Arrays)
- ▶ Insertar/Eliminar (Arrays > BT < listas enlazadas desordenadas)
- ▶ No están limitados ya que conectan sus nodos mediante punteros.
- ▶ **Aplicaciones:**
- ▶ Manipular información jerárquica, hacer fácil la búsqueda de información, para el flujo de trabajo en la composición de imágenes de los efectos visuales creados por programas diseñadores, algoritmos de rutas, toma de decisiones multi-estado en negocios.

Volvamos al código.

- ▶ Ver el árbol básico en:
<https://github.com/JarvisFullStack/DataStructureUcne>

Recorrido de un arbol

- ▶ A diferencia de las listas enlazadas, pilas , colas arrays que tienen un orden lógico para recorrerlos, los arboles poseen varias formas.
- ▶ (a) En-orden (Izq, Raiz, Derech) : 4 2 5 1 3
- ▶ (b) Pre-Orden (Raiz, Izq, Derech) : 1 2 4 5 3
- ▶ (c) Post-Orden (Izq, Derech, Raiz) : 4 5 2 3 1

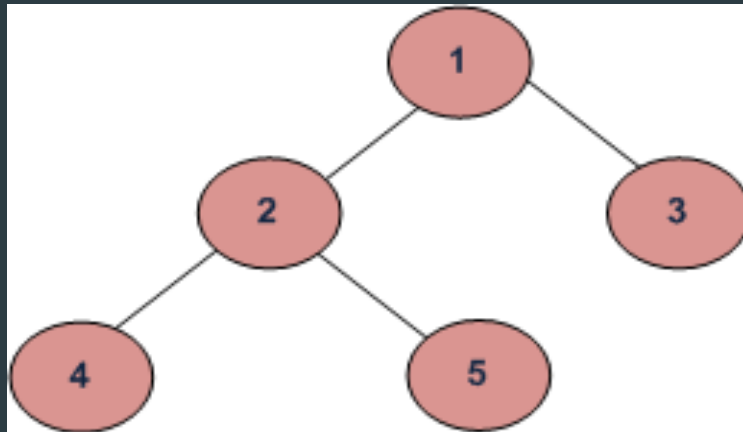


Algoritmos de recorrido (EnOrden)

► En orden

1. Recorrer la parte izquierda del árbol, llamando EnOrden(nodo->izquierda)
2. Visitar la raíz.
3. Recorrer la parte derecha del árbol, llamando EnOrden(nodo->derecha)

4 2 5 1 3

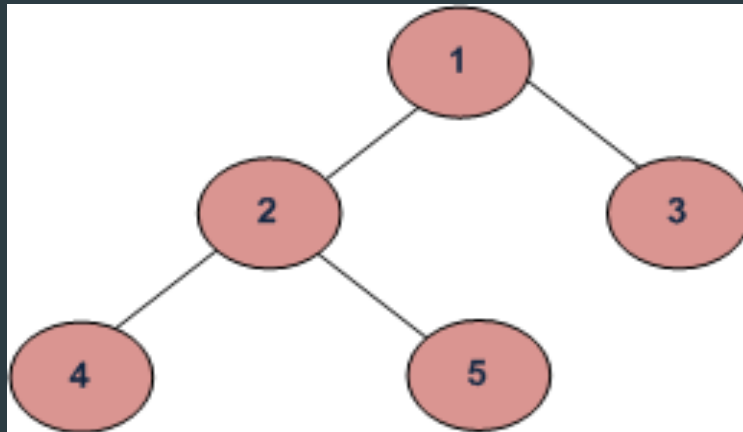


Algoritmos de recorrido (PreOrden)

► PreOrden

1. Visitar la raíz.
2. Recorrer la parte izquierda del árbol, llamando PreOrden(nodo->izquierda)
3. Recorrer la parte derecha del árbol, llamando PreOrden(nodo->derecha)

1 2 4 5 3

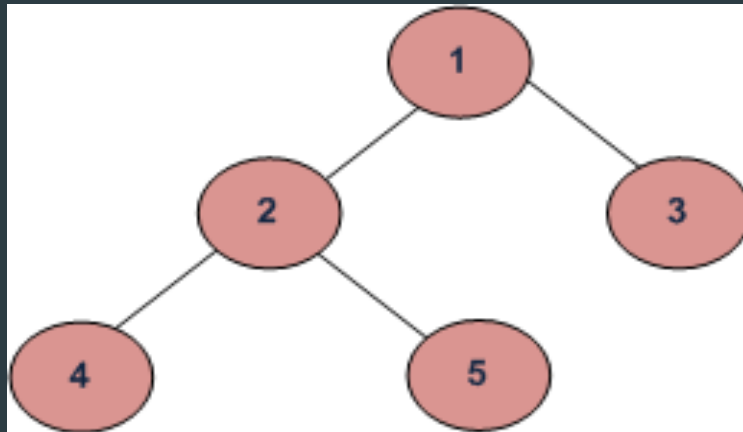


Algoritmos de recorrido (PostOrden)

► PostOrden

1. Recorrer la parte izquierda del árbol, llamando PostOrden(nodo->izquierda)
2. Recorrer la parte derecha del árbol, llamando PostOrden(nodo->derecha)
3. Visitar la raíz.

4 5 2 3 1



Función para contar los nodos hojas

Algoritmo

- ContarNodosHojas(Nodo nodo);
- 1. Si el nodo es nulo retornar cero 0.
- 2. En caso contrario, si izquierda y derecha son nulos retornar 1.
- 3. En caso contrario, calcular de manera recursiva los nodos hojas con la siguiente formula
contador general = conteo de subárboles izquierdo + conteo de subárboles derecho.

Vamos al código. 3

<https://github.com/JarvisFullStack/DataStructureUcne>

