



# Práctica 2

## Unidad: 3

*Asignatura: Estructura  
de datos*

# ARBOLES

**Alumno:** Juan Manuel Kau Sima

**Profesor:** Ruth Martínez

Universidad Tecnológica Metropolitana

**Fecha:** 18/11/2024

**Cuatrimestre y Grupo:** 4-F

## **Instrucción (TAREA EN CLASE)**

### **Fórmulas de los recorridos:**

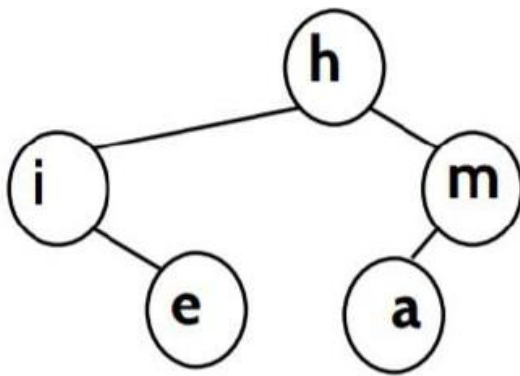
Amplitud: Iniciando desde la raíz, de izquierda a derecha cada nivel del árbol

Preorden: Raíz, Izquierda, Derecha

Postorden: Izquierda, Derecha, Raíz

Inorden: Izquierda, Raíz, Derecha

### **Hacer los recorridos de los siguientes árboles:**



Amplitud: h, i, m, e, a

Inorden (Izquierda, Raíz, Derecha):

En inorden, visitamos el subárbol izquierdo, luego la raíz y finalmente el subárbol derecho. Empezamos por h:

- En el subárbol izquierdo de h, vamos al subárbol izquierdo de i. Allí visitamos el subárbol izquierdo de i (e), luego la raíz i, y finalmente el subárbol derecho de i (a).
- Volvemos a la raíz principal h.
- En el subárbol derecho de h, visitamos su único nodo hijo (m).

Resultado: i, e, h, a, m

Preorden (Raíz, Izquierda, Derecha):

En preorden, visitamos primero la raíz, luego el subárbol izquierdo y finalmente el subárbol derecho. Empezamos por h:

- Desde la raíz h, vamos al subárbol izquierdo. Allí visitamos la raíz i, luego el subárbol izquierdo de i (e), y después el subárbol derecho de i (a).
- Volvemos a la raíz principal h y pasamos al subárbol derecho. Allí visitamos su único nodo hijo (m).

**Resultado: h, i, e, m, a**

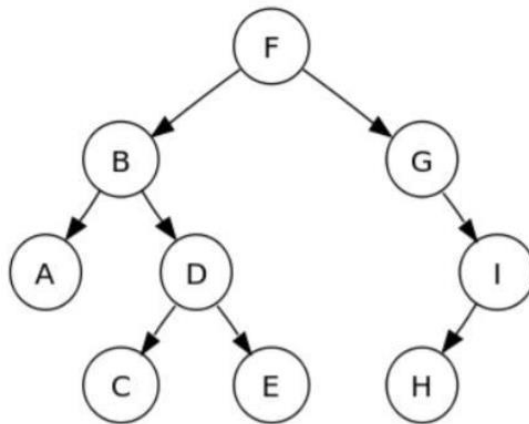
**Postorden (Izquierda, Derecha, Raíz):**

En postorden, visitamos primero el subárbol izquierdo, luego el derecho y finalmente la raíz. Empezamos por h:

- En el subárbol izquierdo de h, comenzamos con el subárbol izquierdo de i (e), luego el subárbol derecho de i (a), y finalmente la raíz i.
- En el subárbol derecho de h, visitamos su único nodo hijo (m).
- Finalmente, regresamos a la raíz principal h.

**Resultado: e, i, a, m, h**

## Árbol 2



**Amplitud:** F, B, G, A, D, I, C, E, H

**Inorden (Izquierda, Raíz, Derecha):**

En inorden, visitamos el subárbol izquierdo, luego la raíz y finalmente el subárbol derecho. Empezamos por F:

- En el subárbol izquierdo de F, vamos al subárbol izquierdo de B. Allí visitamos el subárbol izquierdo de A (no tiene hijos), luego la raíz B, y después el subárbol derecho de B. En este último, visitamos el subárbol izquierdo de D (C), luego la raíz D, y después el subárbol derecho de D (E).
- Volvemos a la raíz principal F.
- En el subárbol derecho de F, vamos al subárbol izquierdo de G (I), luego la raíz G, y finalmente el subárbol derecho de G (H).

**Resultado:** A, B, C, D, E, F, G, H, I

**Preorden (Raíz, Izquierda, Derecha):**

En preorden, visitamos primero la raíz, luego el subárbol izquierdo y finalmente el subárbol derecho. Empezamos por F:

- Desde la raíz F, vamos al subárbol izquierdo. Allí visitamos la raíz B, luego su subárbol izquierdo (A). Pasamos al subárbol derecho de B, donde visitamos la raíz D, luego el subárbol izquierdo de D (C), y finalmente el subárbol derecho de D (E).

- Volvemos a la raíz principal F y pasamos al subárbol derecho. Allí visitamos la raíz G, luego su subárbol izquierdo (I), y finalmente su subárbol derecho (H).

**Resultado: F, B, A, D, C, E, G, I, H**

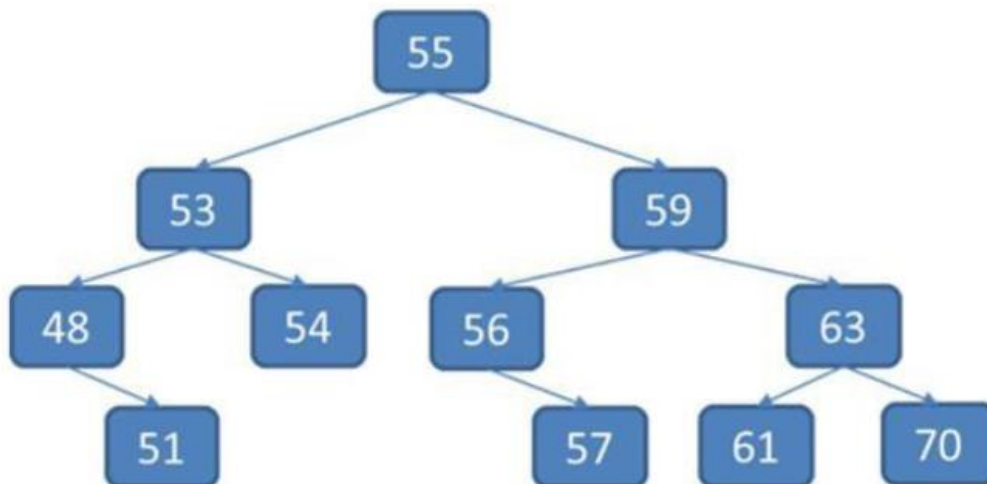
**Postorden (Izquierda, Derecha, Raíz):**

**En postorden, visitamos primero el subárbol izquierdo, luego el derecho y finalmente la raíz. Empezamos por F:**

- En el subárbol izquierdo de F, comenzamos con el subárbol izquierdo de B (A), luego pasamos al subárbol derecho de B. Allí visitamos el subárbol izquierdo de D (C), luego el subárbol derecho de D (E), y finalmente la raíz D. Cerramos con la raíz B.
- En el subárbol derecho de F, comenzamos con el subárbol izquierdo de G (I), luego el subárbol derecho de G (H), y finalmente la raíz G.
- Cerramos con la raíz principal F.

**Resultado: A, C, E, D, B, H, I, G, F**

### Árbol 3:



**Amplitud:** 55, 53, 59, 48, 54, 56, 63, 51, 57, 61, 70

### Inorden (Izquierda, Raíz, Derecha):

En inorden, visitamos el subárbol izquierdo, luego la raíz y finalmente el subárbol derecho. Empezamos por 55:

- En el subárbol izquierdo de 55, vamos al subárbol izquierdo de 53. Allí visitamos el subárbol izquierdo de 48 (vacío), luego la raíz 48, y después su subárbol derecho (51). Volvemos a 53, visitamos su subárbol derecho (54), y finalmente la raíz 53.
- Volvemos a la raíz principal 55.
- En el subárbol derecho de 55, vamos al subárbol izquierdo de 59. Allí visitamos el subárbol izquierdo de 56 (vacío), luego la raíz 56, y después su subárbol derecho (57). Volvemos a 59 y visitamos su subárbol derecho, donde encontramos el subárbol izquierdo de 63 (61), luego la raíz 63, y finalmente su subárbol derecho (70).

**Resultado:** 48, 51, 53, 54, 55, 56, 57, 59, 61, 63, 70

**Preorden (Raíz, Izquierda, Derecha):**

**En preorden, visitamos primero la raíz, luego el subárbol izquierdo y finalmente el subárbol derecho. Empezamos por 55:**

- Desde la raíz 55, vamos al subárbol izquierdo. Allí visitamos la raíz 53, luego su subárbol izquierdo (48), y el subárbol derecho de 48 (51). Volvemos a 53 y visitamos su subárbol derecho (54).
- Volvemos a la raíz principal 55 y pasamos al subárbol derecho. Allí visitamos la raíz 59, luego su subárbol izquierdo (56) y el subárbol derecho de 56 (57). Después, pasamos al subárbol derecho de 59, visitando la raíz 63, su subárbol izquierdo (61), y finalmente su subárbol derecho (70).

**Resultado: 55, 53, 48, 51, 54, 59, 56, 57, 63, 61, 70**

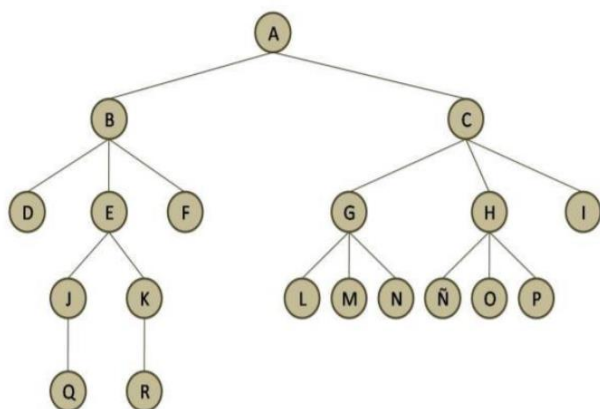
**Postorden (Izquierda, Derecha, Raíz):**

**En postorden, visitamos primero el subárbol izquierdo, luego el derecho y finalmente la raíz. Empezamos por 55:**

- En el subárbol izquierdo de 55, comenzamos con el subárbol izquierdo de 53. En el subárbol izquierdo de 48, no hay más hijos, así que visitamos primero el subárbol derecho de 48 (51), y luego la raíz 48. Después, visitamos el subárbol derecho de 53 (54) y finalmente la raíz 53.
- En el subárbol derecho de 55, comenzamos con el subárbol izquierdo de 59. Allí visitamos el subárbol izquierdo de 56 (vacío), luego el subárbol derecho de 56 (57), y finalmente la raíz 56. En el subárbol derecho de 59, visitamos primero el subárbol izquierdo de 63 (61), luego el subárbol derecho de 63 (70), y finalmente la raíz 63. Cerramos con la raíz 59.
- Finalmente, visitamos la raíz principal 55.

**Resultado: 51, 48, 54, 53, 57, 56, 61, 70, 63, 59, 55**

#### Árbol 4



**Recorrido en amplitud:** A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P, Q, R

**Inorden (Izquierda, Raíz, Derecha):**

En inorden, visitamos el subárbol izquierdo, luego la raíz y finalmente el subárbol derecho. Empezamos por A:

- En el subárbol izquierdo de A, vamos al subárbol izquierdo de B. Allí visitamos D. Luego pasamos al subárbol central de B, recorriendo el subárbol izquierdo de E (llegamos a J), visitamos la raíz E, y después el subárbol derecho de E (llegamos a K). Finalmente, visitamos el subárbol derecho de B, que es F.
- Volvemos a la raíz principal A.
- Pasamos al subárbol derecho de A, comenzando con el subárbol izquierdo de C. Allí encontramos a los hijos de G: L, M, y N (en ese orden). Después de recorrer G, vamos al subárbol central de C y visitamos sus hijos: Ñ y O. Finalmente, recorremos el subárbol derecho de C, que es I.

**Resultado:** D, B, J, E, K, F, A, L, G, M, N, C, Ñ, H, O, P, I

**Preorden (Raíz, Izquierda, Derecha):**

En preorden, visitamos primero la raíz, luego el subárbol izquierdo y finalmente el subárbol derecho. Empezamos por A:

- Desde la raíz A, vamos al subárbol izquierdo y comenzamos con la raíz B. En el subárbol izquierdo de B, visitamos D. Luego vamos al subárbol



central de B, donde encontramos la raíz E, y visitamos primero su subárbol izquierdo (J), luego su subárbol derecho (K). Finalmente, recorremos el subárbol derecho de B, que es F.

- Volvemos a la raíz A y pasamos al subárbol derecho, comenzando con la raíz C. En el subárbol izquierdo de C, visitamos primero la raíz G y sus hijos: L, M, y N. Luego pasamos al subárbol central de C, donde visitamos la raíz H y sus hijos: Ñ y O. Finalmente, recorremos el subárbol derecho de C, que es I.

**Resultado:** A, B, D, E, J, Q, K, R, F, C, G, L, M, N, H, Ñ, O, P, I

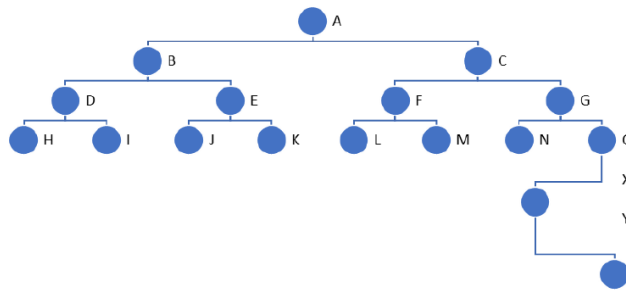
**Postorden (Izquierda, Derecha, Raíz):**

En postorden, visitamos primero el subárbol izquierdo, luego el derecho y finalmente la raíz. Empezamos por A:

- En el subárbol izquierdo de A, comenzamos con el subárbol izquierdo de B, donde visitamos D. Luego pasamos al subárbol central de B, recorriendo primero el subárbol izquierdo de E (J), luego el subárbol derecho de E (K), y finalmente la raíz E. Después de esto, visitamos el subárbol derecho de B (F) y finalmente la raíz B.
- En el subárbol derecho de A, comenzamos con el subárbol izquierdo de C, donde visitamos primero a los hijos de G (L, M, y N), luego la raíz G. Pasamos al subárbol central de C, donde visitamos los hijos de H (Ñ y O), y finalmente la raíz H. Después, visitamos el subárbol derecho de C, que es I, y finalmente la raíz C.
- Terminamos con la raíz principal A.

**Resultado:** D, Q, J, R, K, E, F, B, L, M, N, G, Ñ, O, P, H, I, C, A

## Árbol 5



**Recorrido en amplitud:** A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, X, Y

**Inorden (Izquierda, Raíz, Derecha):**

Para recorrer en inorden, primero visitamos el subárbol izquierdo, luego la raíz y finalmente el subárbol derecho. Empezamos por el nodo A:

- En el subárbol izquierdo de A, vamos al subárbol izquierdo de B. Allí visitamos el subárbol izquierdo de D (llegamos a H), luego la raíz D y después el subárbol derecho de D (llegamos a I). Regresamos a la raíz B, luego visitamos el subárbol derecho de B, donde recorremos el subárbol izquierdo de E (llegamos a J), la raíz E, y el subárbol derecho de E (llegamos a K).
- Volvemos a la raíz principal A.
- Pasamos al subárbol derecho de A, visitando primero el subárbol izquierdo de C. Allí recorremos el subárbol izquierdo de F (llegamos a L), la raíz F, y el subárbol derecho de F (llegamos a M). Luego visitamos la raíz C y su subárbol derecho, donde encontramos el subárbol izquierdo de G (llegamos a N), el subárbol derecho de N (llegamos a O), y continuamos con el subárbol derecho de O (llegamos a X y finalmente Y). Después de esto, regresamos a G.

**Resultado:** H, D, I, B, J, E, K, A, L, F, M, C, N, G, X, O, Y

### **Preorden (Raíz, Izquierda, Derecha):**

**En preorden, visitamos primero la raíz, luego el subárbol izquierdo y finalmente el subárbol derecho. Empezamos por A:**

- Desde la raíz A, vamos al subárbol izquierdo, comenzando con la raíz B. En el subárbol izquierdo de B, visitamos la raíz D, luego el subárbol izquierdo de D (llegamos a H) y el subárbol derecho de D (llegamos a I). Volvemos a B y pasamos al subárbol derecho de B, donde visitamos la raíz E, luego el subárbol izquierdo de E (llegamos a J) y el subárbol derecho de E (llegamos a K).
- Regresamos a la raíz A y pasamos al subárbol derecho, comenzando con la raíz C. En el subárbol izquierdo de C, visitamos la raíz F, luego el subárbol izquierdo de F (llegamos a L) y el subárbol derecho de F (llegamos a M). Después, pasamos al subárbol derecho de C, visitando la raíz G. En el subárbol izquierdo de G, visitamos la raíz N y su subárbol derecho (llegamos a O). Continuamos con el subárbol derecho de O (llegamos a X y finalmente a Y).

**Resultado: A, B, D, H, I, E, J, K, C, F, L, M, G, N, O, X, Y**

### **Postorden (Izquierda, Derecha, Raíz):**

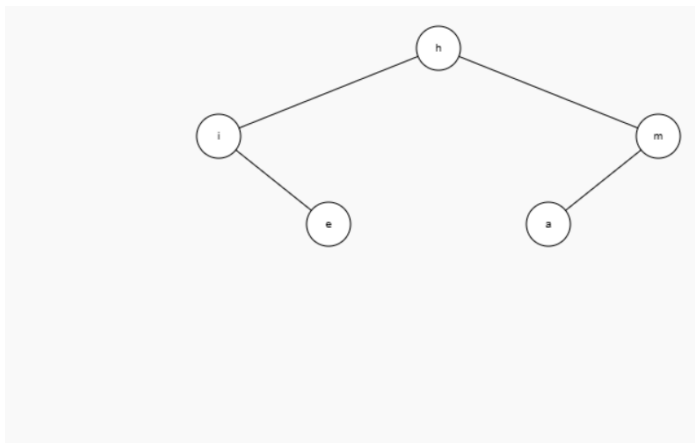
**En postorden, visitamos primero el subárbol izquierdo, luego el derecho y finalmente la raíz. Empezamos por A:**

- En el subárbol izquierdo de A, comenzamos con el subárbol izquierdo de B. En el subárbol izquierdo de D, llegamos a H. Luego visitamos el subárbol derecho de D (llegamos a I) y finalmente la raíz D. Pasamos al subárbol derecho de B, donde en el subárbol izquierdo de E visitamos J, luego el subárbol derecho de E (llegamos a K) y finalmente la raíz E. Terminamos con la raíz B.
- Pasamos al subárbol derecho de A, comenzando con el subárbol izquierdo de C. En el subárbol izquierdo de F, llegamos a L, luego visitamos el subárbol derecho de F (llegamos a M) y finalmente la raíz F. En el subárbol derecho de C, visitamos el subárbol izquierdo de G, donde en el subárbol derecho de N visitamos el subárbol derecho de O (llegamos a X y luego a Y), después regresamos a la raíz O, luego a la raíz N, y finalmente a la raíz G. Cerramos con la raíz C y finalmente la raíz principal A.

**Resultado: H, I, D, J, K, E, B, L, M, F, N, Y, X, O, G, C, A**

## Tarea 2: Programa que compruebe esos recorridos

### Árbol 1:



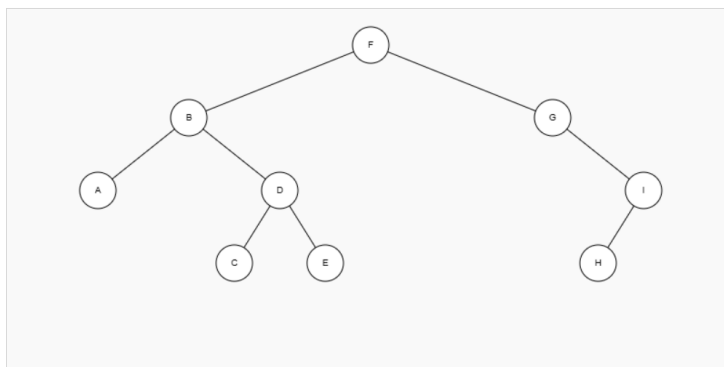
Amplitud: h, i, m, e, a

Preorden: h, i, e, m, a

Inorden: i, e, h, a, m

Postorden: e, i, a, m, h

### Árbol 2:



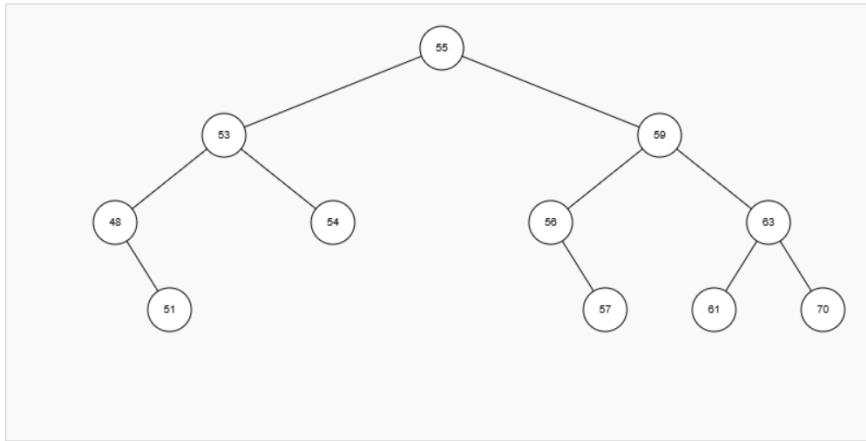
Amplitud: F, B, G, A, D, I, C, E, H

Preorden: F, B, A, D, C, E, G, I, H

Inorden: A, B, C, D, E, F, G, H, I

Postorden: A, C, E, D, B, H, I, G, F

### Árbol 3:



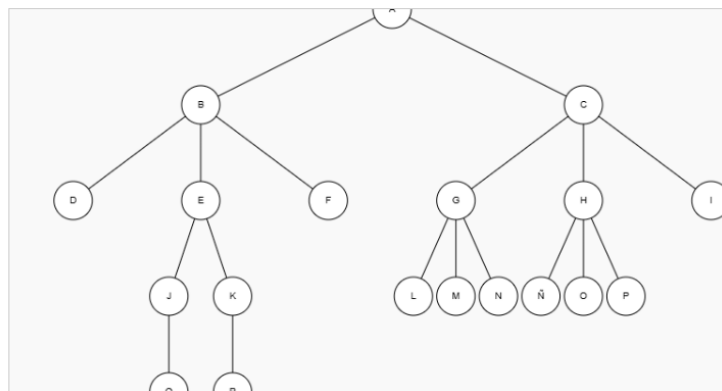
Amplitud: 55, 53, 59, 48, 54, 56, 63, 51, 57, 61, 70

Preorden: 55, 53, 48, 51, 54, 59, 56, 57, 63, 61, 70

Inorden: 48, 51, 53, 54, 55, 56, 57, 59, 61, 63, 70

Postorden: 51, 48, 54, 53, 57, 56, 61, 70, 63, 59, 55

### Árbol 4:



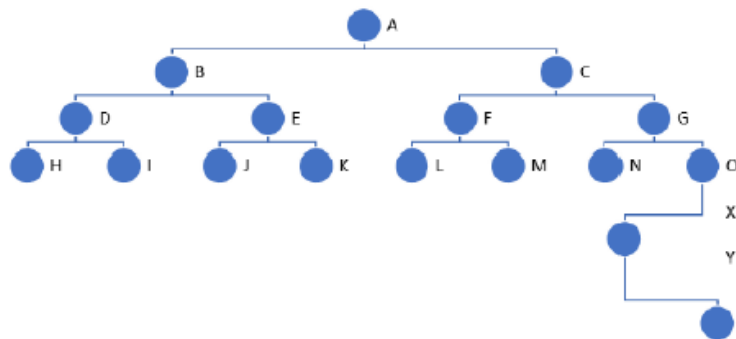
Amplitud: A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P, Q, R

Preorden: A, B, D, E, J, Q, K, R, F, C, G, L, M, N, H, Ñ, O, P, I

Inorden: D, B, J, Q, E, K, R, F, A, L, G, M, N, C, Ñ, H, O, P, I

Postorden: D, Q, J, R, K, E, F, B, L, M, N, G, Ñ, O, P, H, I, C, A

## Árbol 5:



Amplitud: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, X, Y

Preorden: A, B, D, H, I, E, J, K, C, F, L, M, G, N, O, X, Y

Inorden: H, D, I, B, J, E, K, A, L, F, M, C, N, G, X, Y, O

Postorden: H, I, D, J, K, E, B, L, M, F, N, Y, X, O, G, C, A

## Script.js:

```
1 // Clase para un nodo del árbol
2 class TreeNode {
3   constructor(value) {
4     this.value = value;
5     this.left = null;
6     this.right = null;
7   }
8 }
9
10 // Clase para el Árbol Binario
11 class BinaryTree {
12   constructor() {
13     this.root = null;
14   }
15 }
16
17 // Clase para manejar la interfaz de usuario
18 class TreeApp {
19   constructor() {
20     // Elementos del DOM
21     this.treeSizeInput = document.getElementById('tree-size');
22     this.generateTreeButton = document.getElementById('generate-tree');
23     this.treeInputsDiv = document.getElementById('tree-inputs');
24     this.outputDiv = document.getElementById('output');
25     this.canvas = document.getElementById('treeCanvas');
26     this.ctx = this.canvas.getContext('2d');
27     this.buildTreeButton = document.getElementById('build-tree');
28
29     // Botones de acciones
30     this.bfsButton = document.getElementById('bfs');
31     this.preorderButton = document.getElementById('preorder');
32     this.inorderButton = document.getElementById('inorder');
33     this.postorderButton = document.getElementById('postorder');
34
35     // Instancia del árbol
36     this.tree = new BinaryTree();
37   }
38 }
```

```

38 // Configurar eventos
39 this.setEventListeners();
40 }
41
42 // Configurar eventos
43 setEventListeners() {
44   this.generateTreeButton.addEventListener('click', () => this.generateTreeInputs());
45   this.buildTreeButton.addEventListener('click', () => {
46     const isValid = this.buildTreeFromInputs();
47     if (isValid) this.drawTree();
48   });
49   this.bfsButton.addEventListener('click', () => this.handleTraversal('bfs'));
50   this.preorderButton.addEventListener('click', () => this.handleTraversal('preorder'));
51   this.inorderButton.addEventListener('click', () => this.handleTraversal('inorder'));
52   this.postorderButton.addEventListener('click', () => this.handleTraversal('postorder'));
53 }
54
55 // Generar campos para los valores del árbol
56 generateTreeInputs() {
57   const size = parseInt(this.treeSizeInput.value);
58   if (isNaN(size) || size <= 0) {
59     alert('Por favor, introduce un tamaño válido.');
```

```

60     return;
61   }
62
63   this.treeInputsDiv.innerHTML = ''; // Limpiar inputs anteriores
64   for (let i = 0; i < size; i++) {
65     const container = document.createElement('div');
66     container.classList.add('node-input');
67
68     const input = document.createElement('input');
69     input.type = 'text';
70     input.placeholder = i === 0 ? 'Raíz (obligatoria)' : `Valor Nodo ${i + 1}`;
71     input.classList.add('tree-value');
```

```

72
73     const leftCheckbox = document.createElement('input');
74     leftCheckbox.type = 'checkbox';
75     leftCheckbox.classList.add('has-left');
76     const leftLabel = document.createElement('label');
77     leftLabel.textContent = 'Hijo Izq';
78
79     const rightCheckbox = document.createElement('input');
80     rightCheckbox.type = 'checkbox';
81     rightCheckbox.classList.add('has-right');
82     const rightLabel = document.createElement('label');
83     rightLabel.textContent = 'Hijo Der';
84
85     container.appendChild(input);
86     container.appendChild(leftCheckbox);
87     container.appendChild(leftLabel);
88     container.appendChild(rightCheckbox);
89     container.appendChild(rightLabel);
90     this.treeInputsDiv.appendChild(container);
91   }
92 }
93
94 // Construir el árbol desde los inputs definidos por el usuario
95 buildTreeFromInputs() {
96   const inputs = document.querySelectorAll('.node-input');
97   if (inputs.length === 0) {
98     alert('Por favor, genera entradas primero.');
```

```

99     return false;
100   }

```

```

102 // Crear un array de nodos basado en las entradas
103 const nodes = [];
104 inputs.forEach(input => {
105     const value = input.querySelector('.tree-value').value;
106     if (!value) {
107         nodes.push(null); // Si no hay valor, es un nodo vacío
108     } else {
109         nodes.push({
110             value: value,
111             left: input.querySelector('.has-left').checked,
112             right: input.querySelector('.has-right').checked
113         });
114     }
115 });
116
117 // Validar que el primer nodo (raíz) exista
118 if (!nodes.length || nodes[0] === null) {
119     alert('Por favor, define al menos un nodo raíz.');
120     return false;
121 }
122
123 // Crear la raíz
124 const rootValue = nodes[0].value;
125 this.tree.root = new TreeNode(rootValue);
126
127 // Usar una cola para construir el árbol dinámicamente
128 const queue = [{ node: this.tree.root, data: nodes[0] }];
129 let currentIndex = 1; // Índice para los nodos restantes
130

```

```

131 while (queue.length > 0) {
132     const { node, data } = queue.shift(); // Tomar el nodo actual y su configuración
133
134     // Crear hijo izquierdo solo si la casilla está marcada
135     if (data.left) {
136         if (currentIndex < nodes.length && nodes[currentIndex]) {
137             const leftNodeData = nodes[currentIndex];
138             if (!isNaN(node.value) && !isNaN(leftNodeData.value)) {
139                 // Validar números: izquierdo debe ser menor que el nodo padre
140                 if (parseInt(leftNodeData.value) >= parseInt(node.value)) {
141                     alert('Error: El valor del hijo izquierdo (${leftNodeData.value}) debe ser menor que el nodo padre (${node.value}).');
142                     return false;
143                 }
144             }
145             const leftNode = new TreeNode(leftNodeData.value);
146             node.left = leftNode;
147             queue.push({ node: leftNode, data: leftNodeData });
148         }
149         currentIndex++; // Incrementar índice solo si un hijo izquierdo está marcado
150     }
151
152     // Crear hijo derecho solo si la casilla está marcada
153     if (data.right) {
154         if (currentIndex < nodes.length && nodes[currentIndex]) {
155             const rightNodeData = nodes[currentIndex];
156             if (!isNaN(node.value) && !isNaN(rightNodeData.value)) {
157                 // Validar números: derecho debe ser mayor que el nodo padre
158                 if (parseInt(rightNodeData.value) <= parseInt(node.value)) {
159                     alert('Error: El valor del hijo derecho (${rightNodeData.value}) debe ser mayor que el nodo padre (${node.value}).');
160                     return false;
161                 }
162             }
163             const rightNode = new TreeNode(rightNodeData.value);
164             node.right = rightNode;
165             queue.push({ node: rightNode, data: rightNodeData });
166         }
167     }
168 }

```





```

235     let result = [];
236     switch (type) {
237       case 'bfs':
238         result = this.bfs();
239         this.displayOutput(`Amplitud: ${result.join(', ')}`);
240         break;
241       case 'preorder':
242         result = this.preorder(this.tree.root);
243         this.displayOutput(`Preorden: ${result.join(', ')}`);
244         break;
245       case 'inorder':
246         result = this.inorder(this.tree.root);
247         this.displayOutput(`Inorden: ${result.join(', ')}`);
248         break;
249       case 'postorder':
250         result = this.postorder(this.tree.root);
251         this.displayOutput(`Postorden: ${result.join(', ')}`);
252         break;
253       default:
254         this.displayOutput(`Tipo de recorrido no válido.`);
255         break;
256     }
257   }
258
259   bfs() {
260     const queue = [this.tree.root];
261     const result = [];
262     while (queue.length > 0) {
263       const node = queue.shift();
264       result.push(node.value);
265       if (node.left) queue.push(node.left);
266       if (node.right) queue.push(node.right);
267     }
268     return result;
269   }

```

```

271   preorder(node, result = []) {
272     if (!node) return result;
273     result.push(node.value);
274     this.preorder(node.left, result);
275     this.preorder(node.right, result);
276     return result;
277   }
278
279   inorder(node, result = []) {
280     if (!node) return result; // Si el nodo es null
281
282     // Recorrer el subárbol izquierdo
283     this.inorder(node.left, result);
284
285     // Procesar el nodo actual (raíz)
286     result.push(node.value);
287
288     // Recorrer el subárbol derecho
289     this.inorder(node.right, result);
290
291     return result; // Retornar el resultado
292   }
293
294
295   postorder(node, result = []) {
296     if (!node) return result;
297     this.postorder(node.left, result);
298     this.postorder(node.right, result);
299     result.push(node.value);
300     return result;
301   }
302
303   displayOutput(message) {
304     this.outputDiv.textContent = message;
305   }
306 }

```

```

308     document.getElementById('show-nary-tree').addEventListener('click', () => {
309       window.location.href = 'naryTree.html'; // Redirige a la página del árbol n-ario
310     });
311
312     // Inicializar la aplicación
313     document.addEventListener('DOMContentLoaded', () => {
314       new TreeApp();
315     });

```

**Partes principales del código:**

**Clases principales:**

- **TreeNode:** Representa un nodo de árbol con un valor y punteros a sus hijos (izquierdo y derecho).
- **BinaryTree:** Representa el árbol binario, con una raíz inicializada como null.
- **TreeApp:** Maneja la interfaz de usuario y las operaciones del árbol (construcción, dibujo y recorridos).

**Interfaz gráfica:**

- **Inputs dinámicos:** Permite al usuario especificar el tamaño del árbol y los valores de los nodos.
- **Canvas:** Dibuja el árbol binario en un lienzo con conexiones y nodos circulares.

**Construcción del árbol:**

- Los usuarios ingresan valores de nodos y seleccionan si tienen hijos (izquierdo/derecho).
- Valida que los hijos izquierdos/derecho cumplan con las propiedades del árbol binario (izquierdo < raíz < derecho).

**Recorridos del árbol:**

- Implementa cuatro tipos:
  - **BFS (Recorrido por niveles):** Usa una cola para recorrer el árbol en amplitud.
  - **Preorden:** Nodo → Subárbol izquierdo → Subárbol derecho.
  - **Inorden:** Subárbol izquierdo → Nodo → Subárbol derecho.
  - **Postorden:** Subárbol izquierdo → Subárbol derecho → Nodo.

**Visualización del árbol:**

- Se dibuja usando coordenadas en el canvas.
- Conecta nodos con líneas y ajusta el espaciado dinámicamente.

**Interacción:**

- Botones en la interfaz permiten generar el árbol, dibujarlo y ejecutar distintos recorridos.
- Redirección a otra página para trabajar con árboles n-arios.