



UPN, PASIÓN POR TRANSFORMAR VIDAS

UNIDAD 2: ÁRBOLES Y GRAFOS

SESIÓN 8: ÁRBOLES BINARIOS

Recorrido inorden, postorden, preorden.

Arboles binarios de Búsqueda ABB.

Dr. Eric Gustavo Coronel Castillo
eric.coronel@upn.pe



DESAFIO DEL DIA

"Cada línea de código es un paso hacia una gran solución; las estructuras de datos guían el camino hacia el éxito"



IDEA FUERZA

Programar con intención y estructura
asegura progresos continuos y sólidos.

LOGRO DE LA UNIDAD 1



Al finalizar la unidad, el estudiante implementa algoritmos utilizando árboles y grafos, como resultado del análisis de casos utilizando el lenguaje C# con entorno gráfico, demostrando lógica y habilidad en la implementación de los algoritmos para su proyecto final..

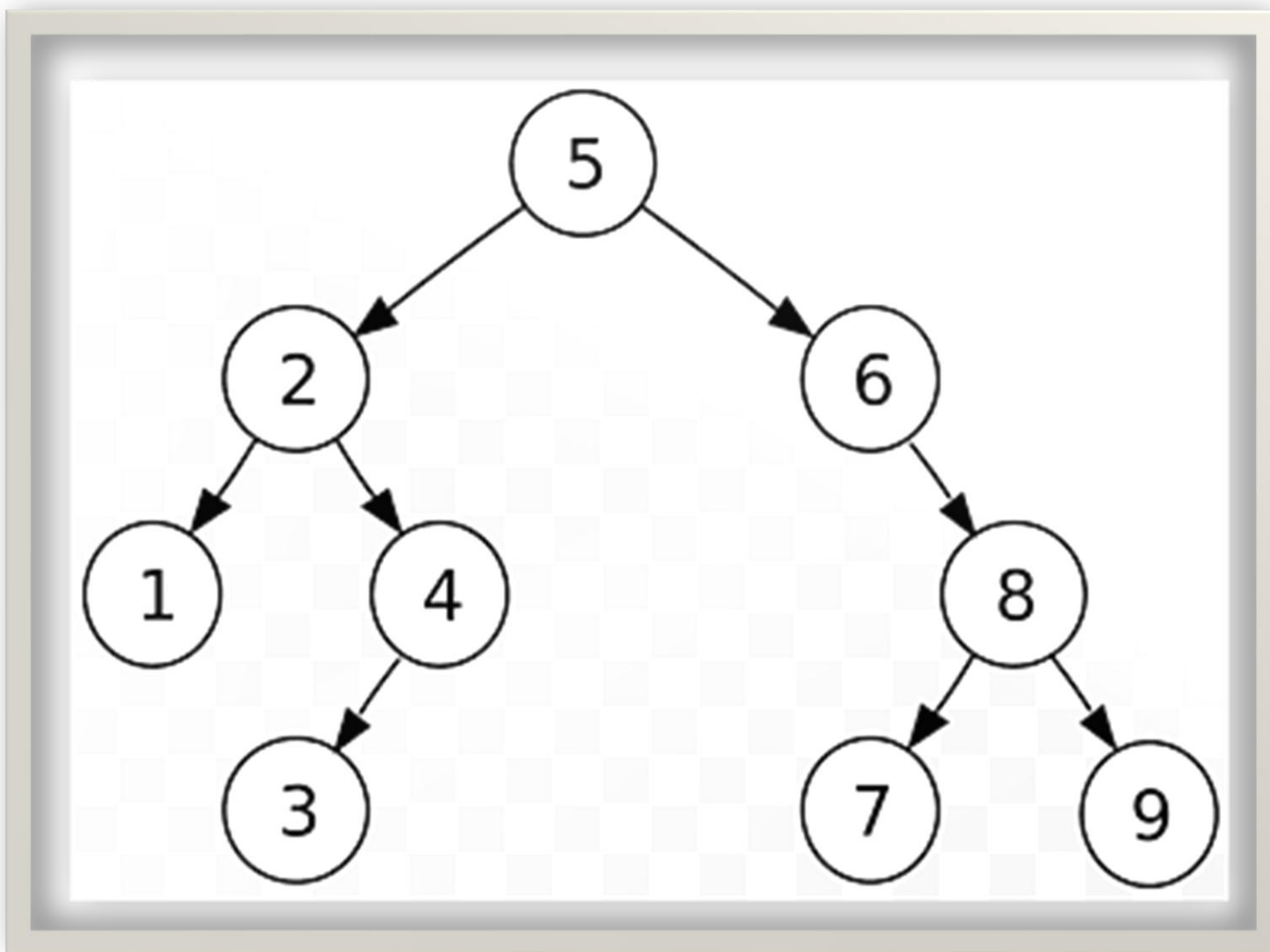


Al término de la sesión, el estudiante aprende algoritmos con arboles, arboles binarios y diversas aplicaciones, usándolos con coherencia.

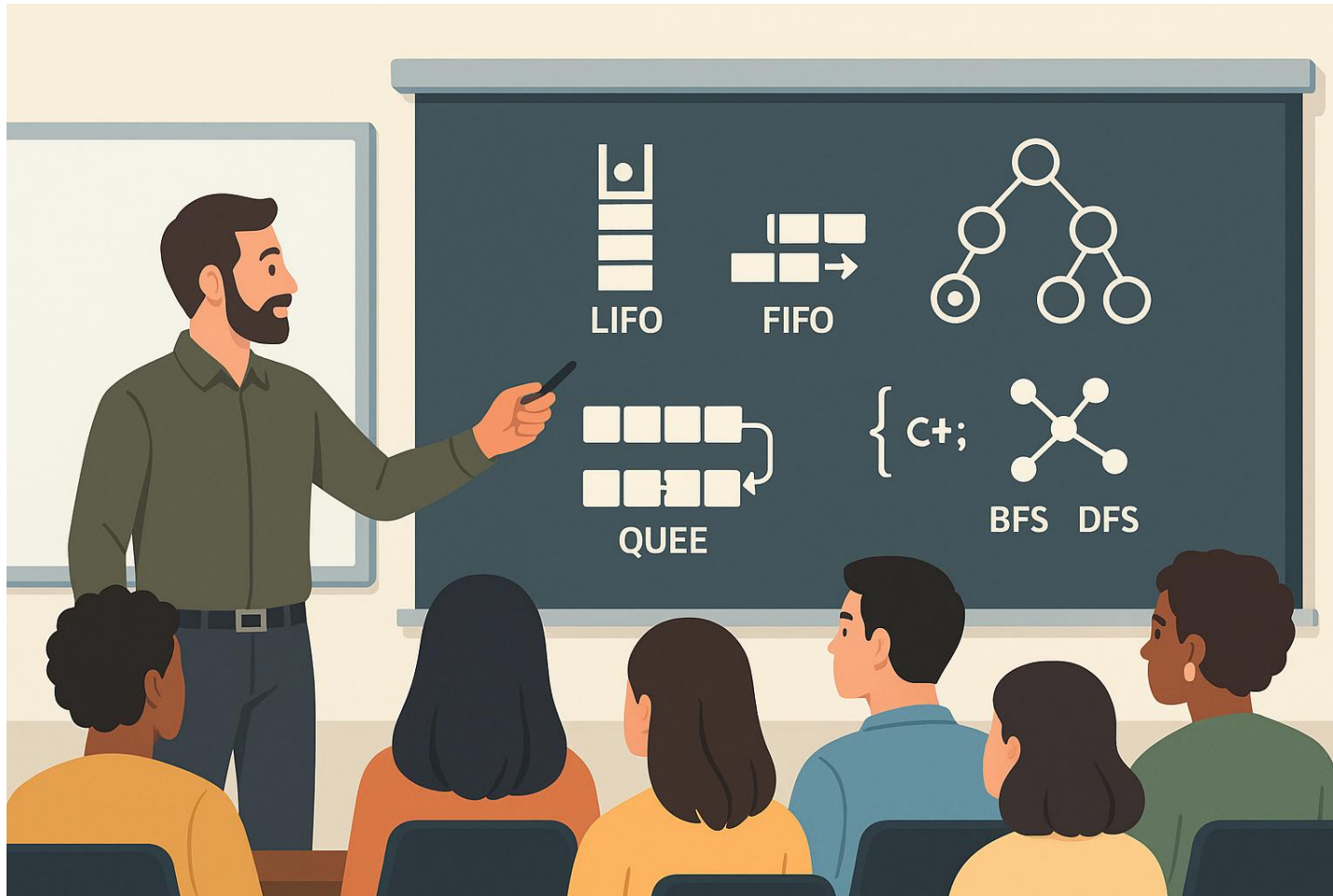
- Recorrido inorden, postorden, preorden.
- Arboles binarios de Búsqueda ABB.



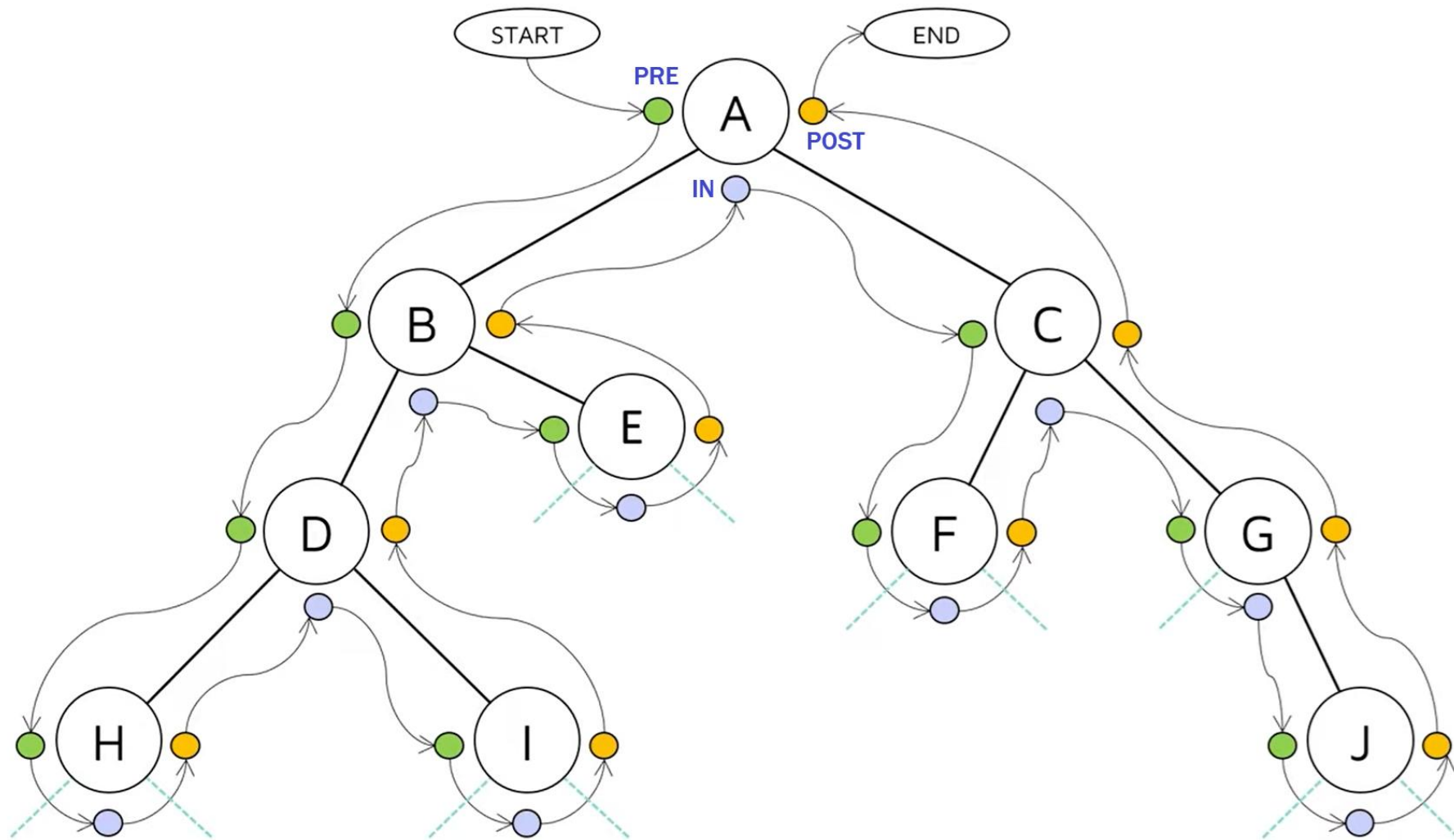
1. ¿Qué es árbol?
2. ¿Qué es un árbol binario?



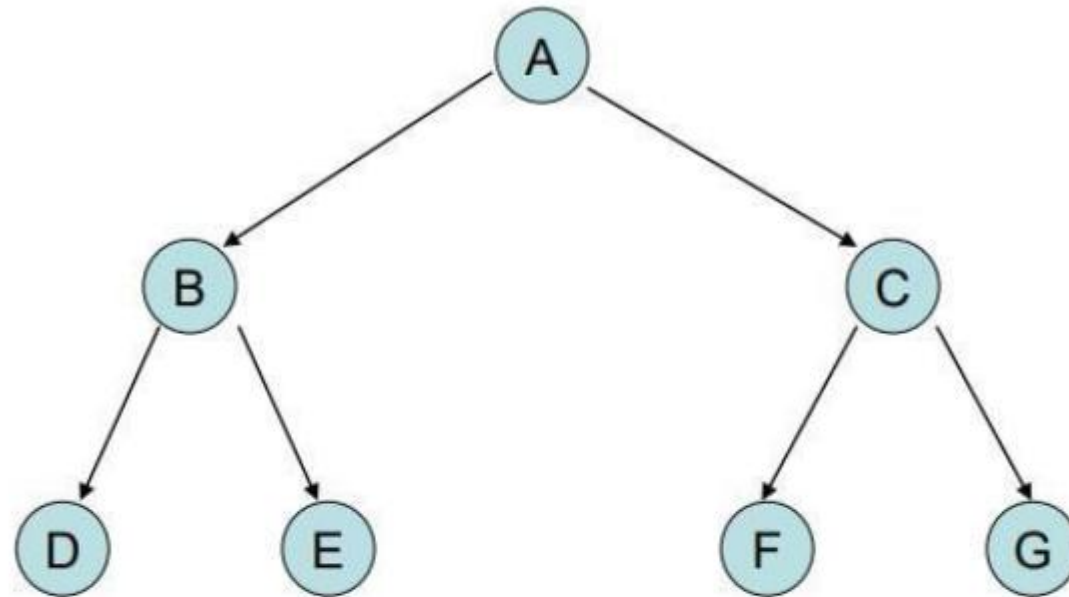
DESARROLLO



ARBOL BINARIO



ARBOL BINARIO



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

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

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

DATOS IMPORTANTES DE LOS ÁRBOLES



Nodos: Se le llama Nodo a cada elemento que contiene un Árbol.

Nodo Raíz: Se refiere al primer nodo de un Árbol, Solo un nodo del Árbol puede ser la Raíz.

Nodo Padre: Se utiliza este termino para llamar a todos aquellos nodos que tiene al menos un hijo.

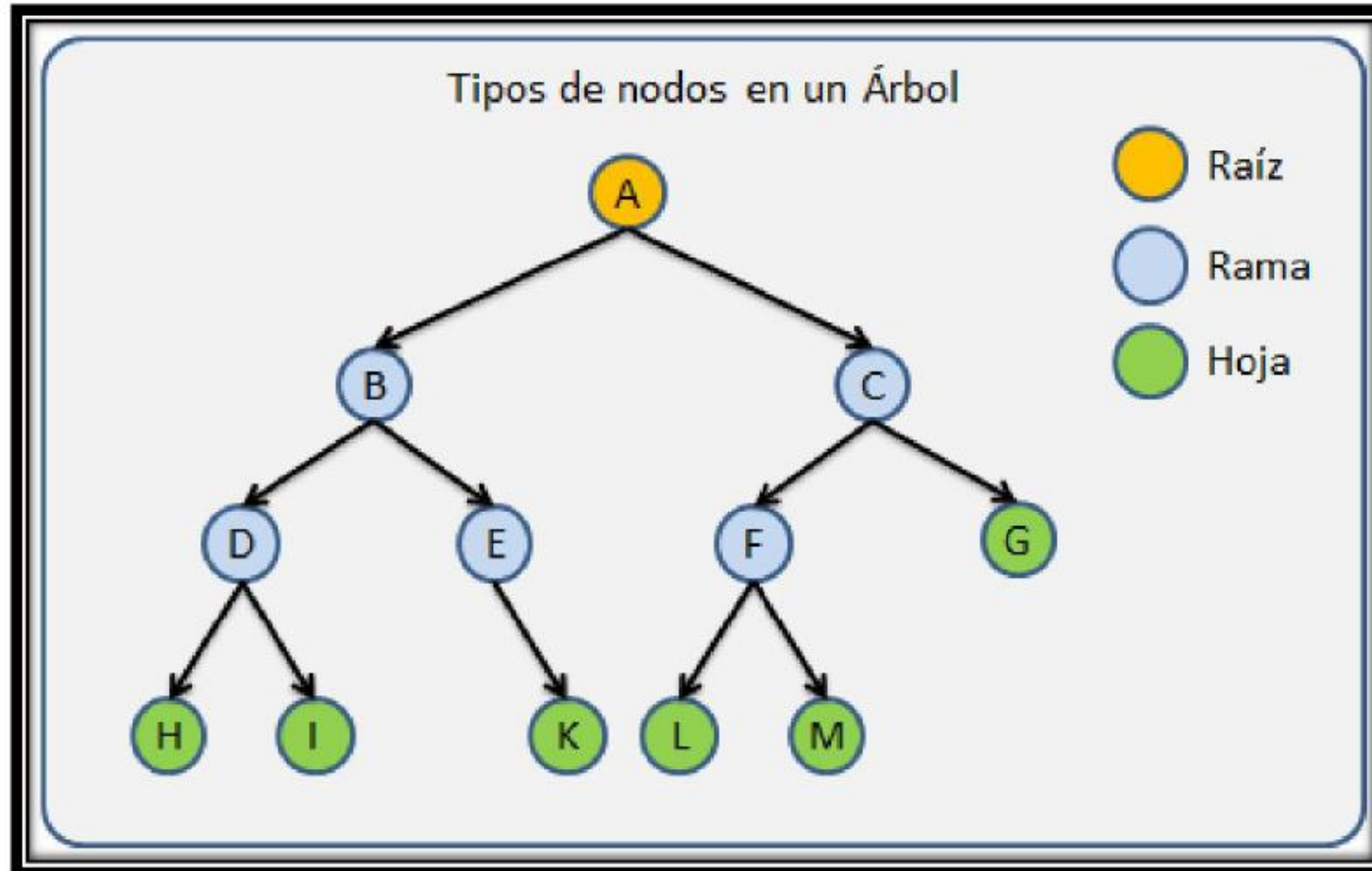
Nodo Hijo: Los hijos son todos aquellos nodos que tiene un padre.

Nodo Hermano: Los nodos hermanos son aquellos nodos que comparte a un mismo padre en común dentro de la estructura.

Nodo Hoja: Son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura.

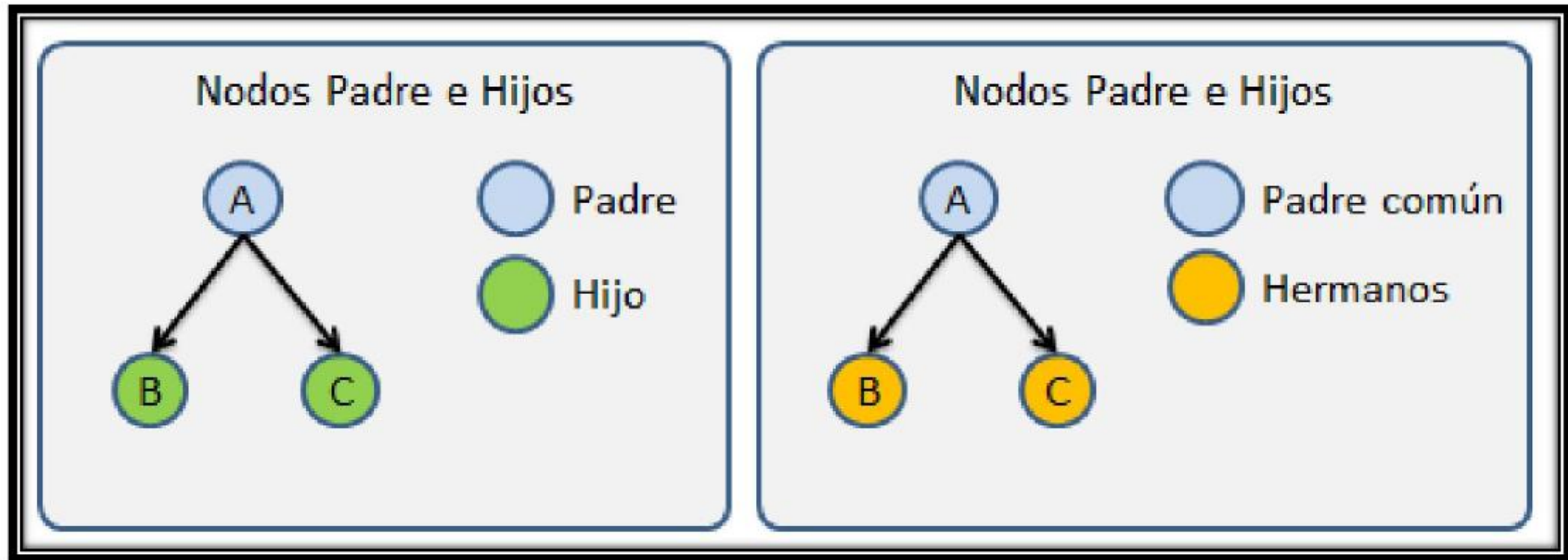
Nodo Rama: Estos son todos aquellos nodos que no son la raíz y que además tiene al menos un hijo.

DATOS IMPORTANTES DE LOS ÁRBOLES



La imagen muestra de forma gráfica cuales son los nodos Raíz, Rama, Hoja.

DATOS IMPORTANTES DE LOS ÁRBOLES




La siguiente imagen muestra de forma gráfica los nodos Padre, Hijo y Hermanos.


ÁRBOL BINARIO



- Preorder:
 - Visitar nodo
 - Visitar árbol izquierdo
 - Visitar árbol derecho
- Inorder:
 - Visitar árbol izquierdo
 - Visitar nodo
 - Visitar árbol derecho
- Postorder:
 - Visitar árbol izquierdo
 - Visitar árbol derecho
 - Visitar nodo



```
void inorder(NodoArbol *nodo) {  
    if (nodo != NULL) {  
        inorder(nodo->izq);  
        visitar(nodo);  
        inorder(nodo->der);  
    }  
}
```

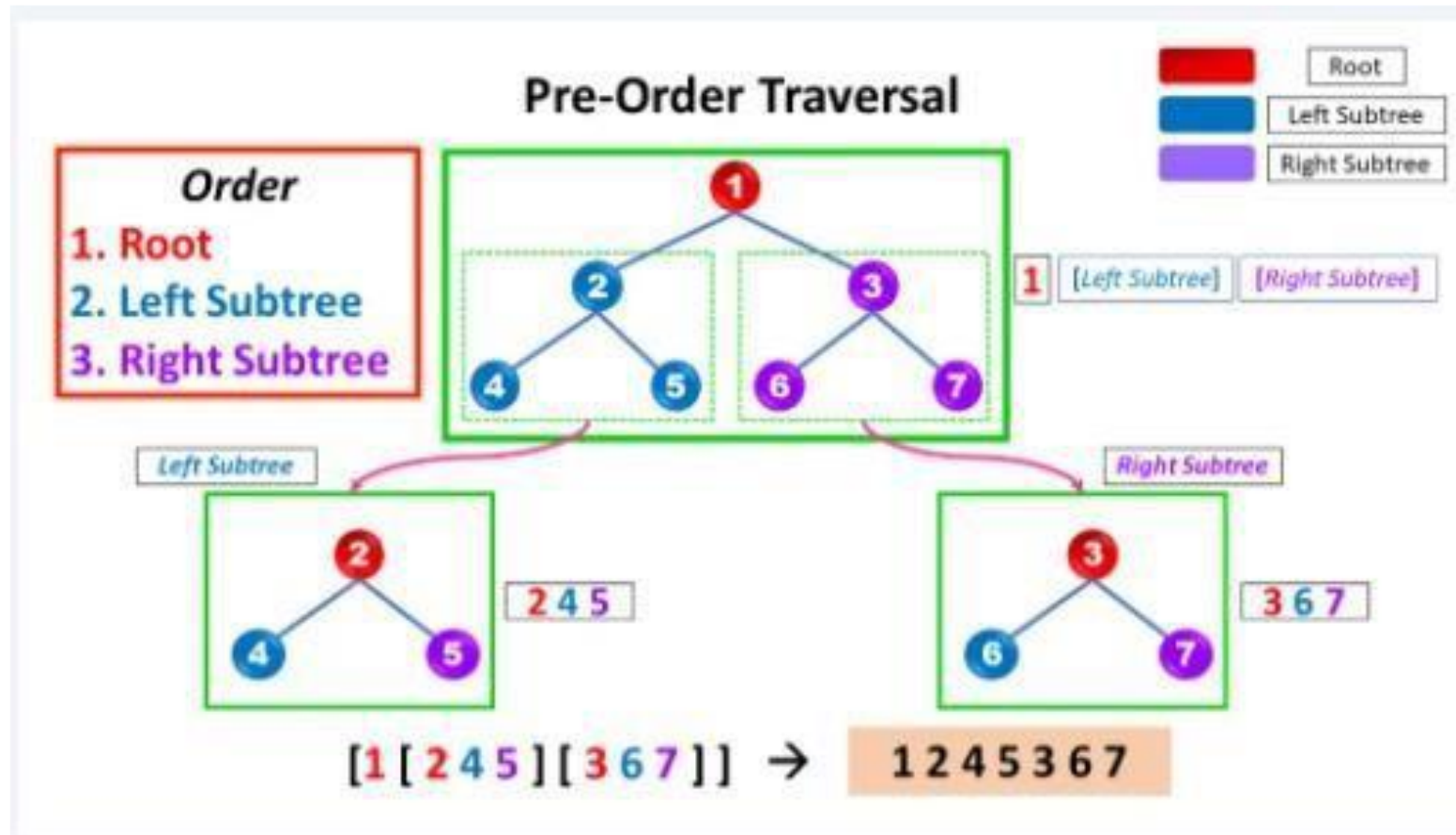


```
void postorder(NodoArbol *nodo)  
{  
    if (nodo != NULL) {  
        postorder(nodo->izq);  
        postorder(nodo->der);  
        visitar(nodo);  
    }  
}
```


ÁRBOL BINARIO



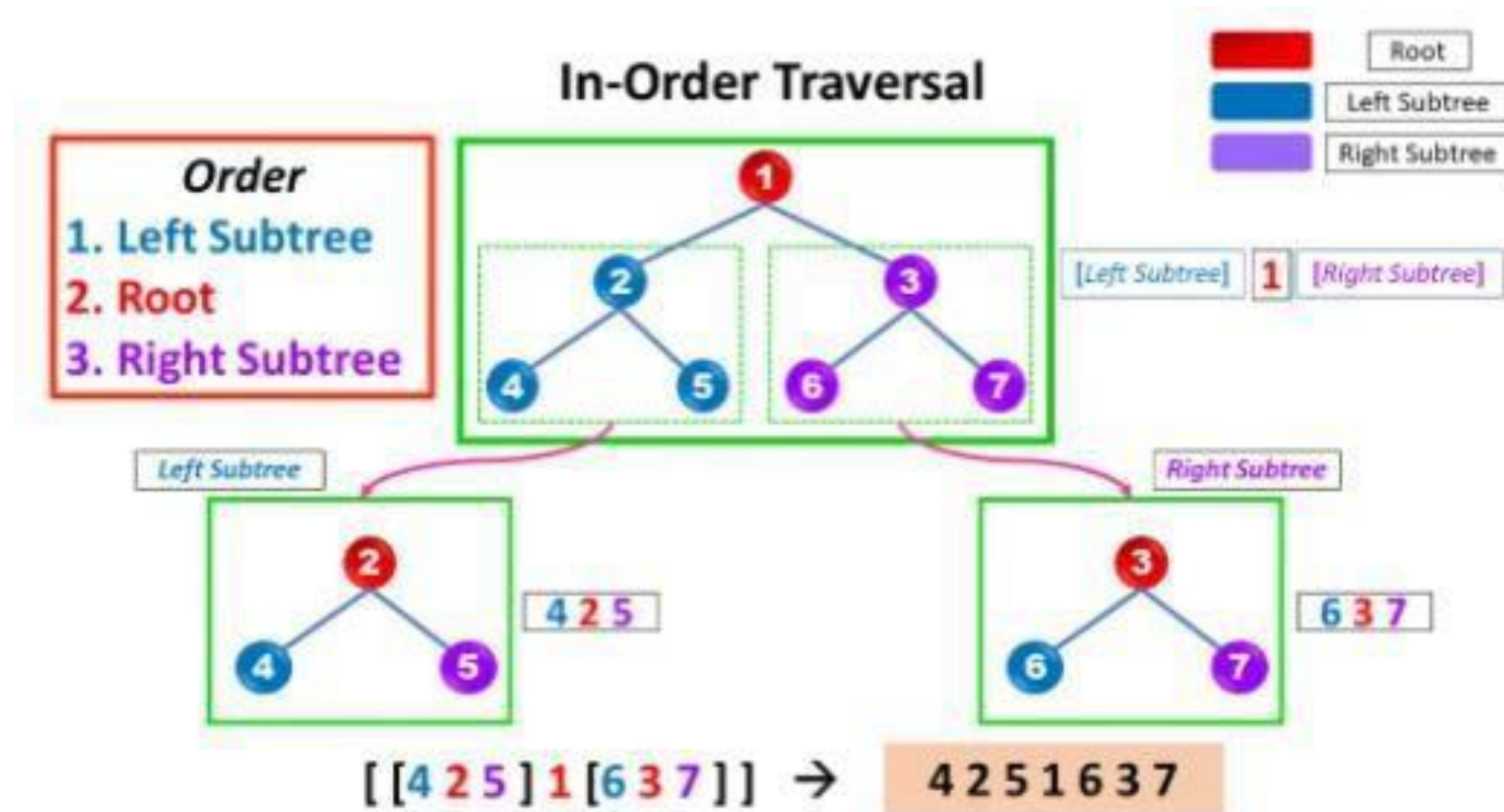
En el recorrido PreOrder, se visita primero el nodo raíz, luego el subárbol izquierdo, y finalmente el subárbol derecho.



ÁRBOL BINARIO



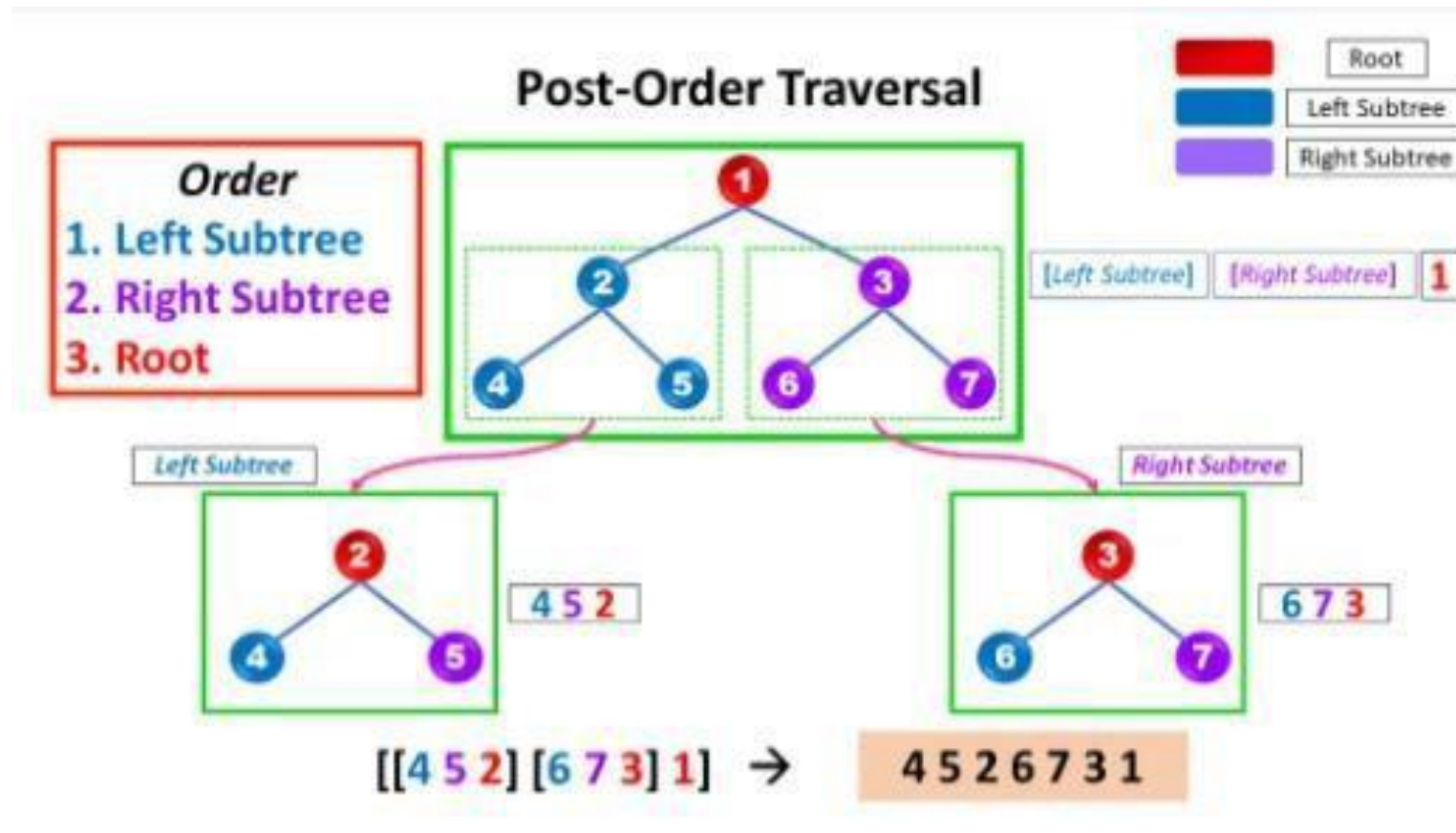
En el recorrido InOrder, primero se visita el subárbol izquierdo, luego el nodo raíz y finalmente el subárbol derecho.



ÁRBOL BINARIO



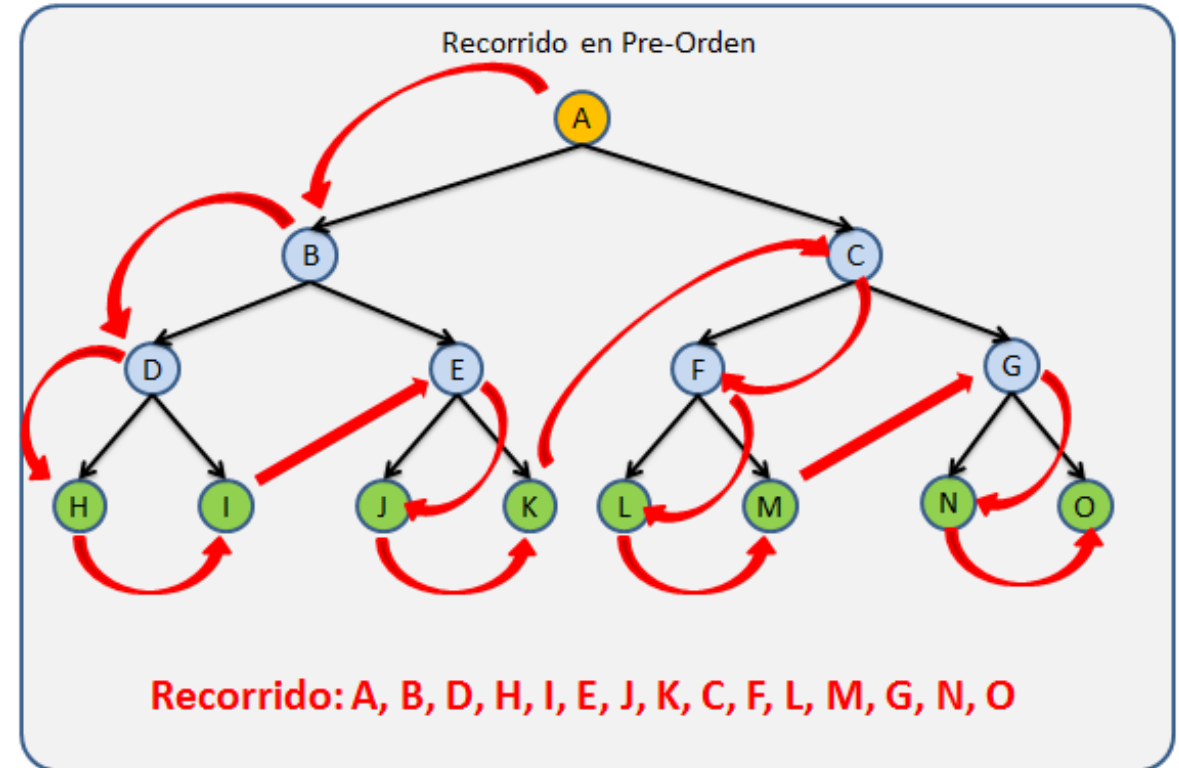
En el recorrido PostOrder, se visita primero el subárbol izquierdo, luego el subárbol derecho y finalmente el nodo raíz.



BÚSQUEDA EN PROFUNDIDAD



Recorrido Pre-orden: El recorrido inicia en la Raíz y luego se recorre en pre-orden cada uno de los subárboles de izquierda a derecha. Esta definición puede ser un poco compleja de entender por lo que mejor les dejo la siguiente imagen.

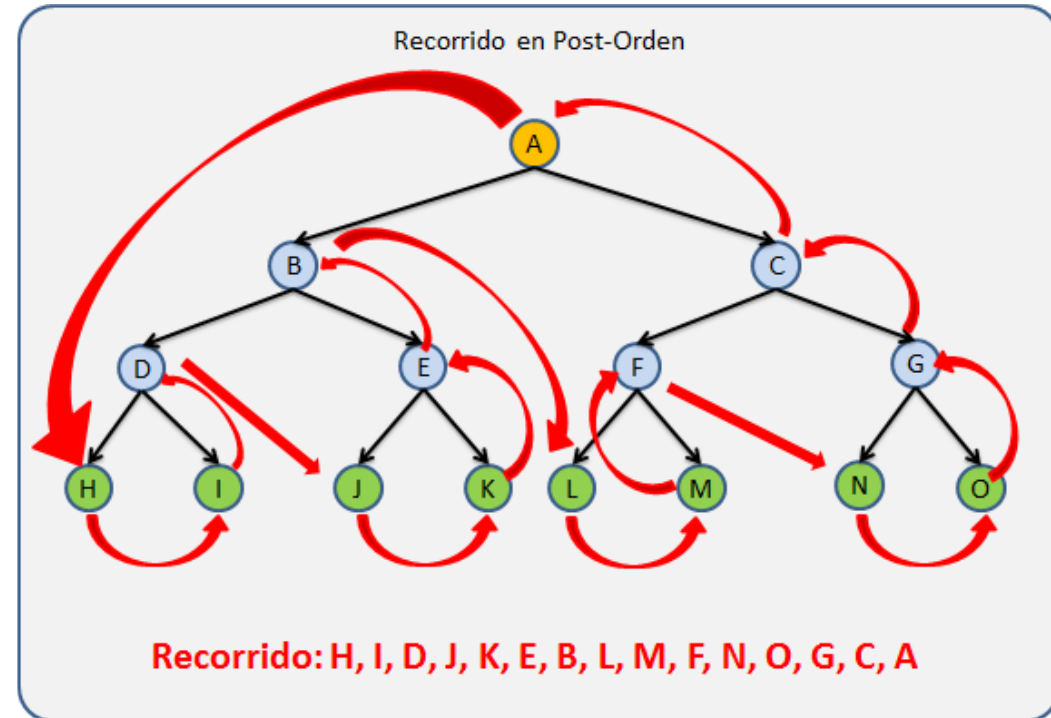


En la imagen podemos ver el orden en que es recorrido el árbol iniciando desde la Raíz.

BÚSQUEDA EN PROFUNDIDAD



Recorrido Pos-orden: Se recorre el pos-orden cada uno de los sub-árboles y al final se recorre la raíz. Para comprender mejor esta definición observemos la siguiente imagen. En la imagen podemos observar como se realiza el recorrido en Pos-Orden, Sin embargo es importante notar que el primer nodo que se imprime no es la Raíz pues en este recorrido la Raíz de cada Sub-Árbol es procesado al final, ya que toda su descendencia ha sido procesada.

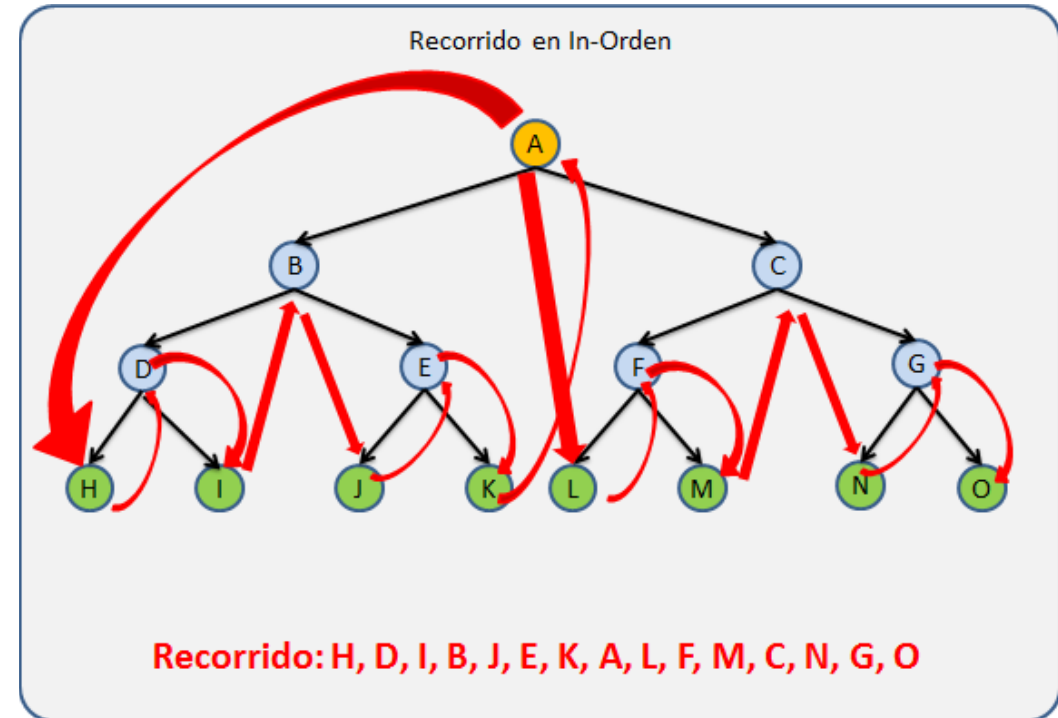


BÚSQUEDA EN PROFUNDIDAD



Recorrido in-orden: Se recorre en in-orden el primer sub-árbol, luego se recorre la raíz y al final se recorre en in-orden los demás sub-árboles.

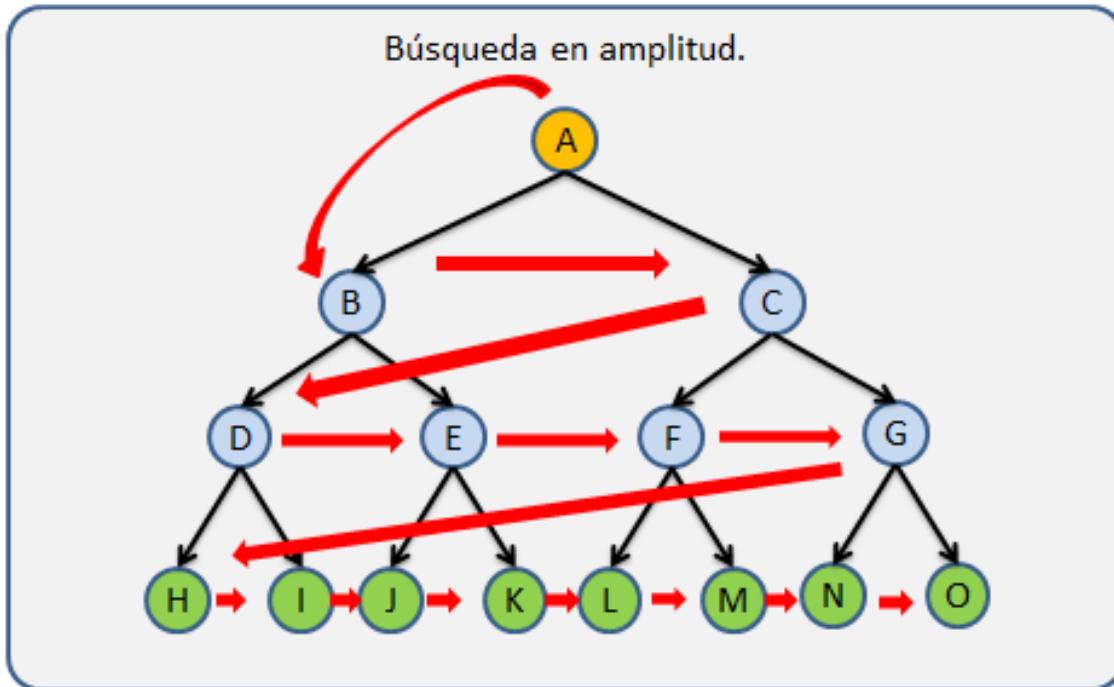
En la imagen se muestra como es el recorrido In-Orden, Podemos apreciar que la Raíz no es el primero elemento en ser impreso pues este recorrido recorre su rama izquierda, luego la raíz del sub-árbol y luego la rama derecha.



BÚSQUEDA EN AMPLITUD



Se recorre primero la raíz, luego se recorren los demás nodos ordenados por el nivel al que pertenecen en orden de Izquierda a derecha. Este tipo de búsqueda se caracteriza por que la búsqueda se hace nivel por nivel y de izquierda a derecha.



En la imagen se observa como es que un nodo es buscado mediante la búsqueda en profundidad.

APLICACIÓN: EVALUACION DE EXPRESIONES

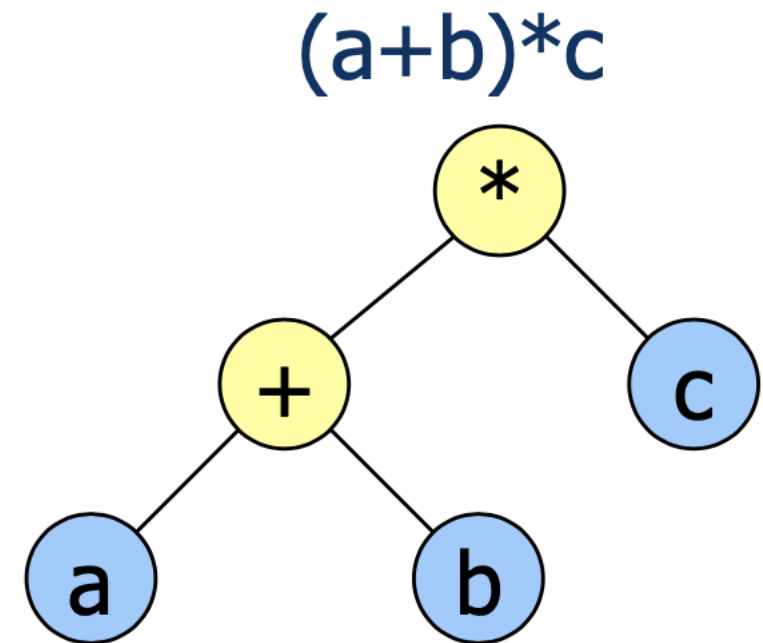
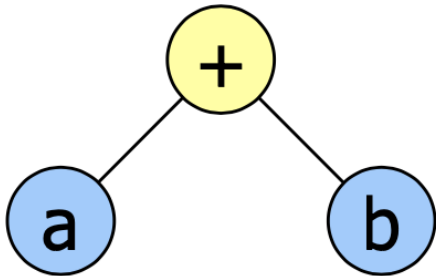


- Ya sabemos lo de las expresiones, cierto?
 - InFija, operador en medio
 - PreFija, operador antes de dos operandos
 - PosFija, operador luego de dos operandos
- Para evaluar una expresion dada, podriamos
 - Pasarla a posfija y usar solo pilas
 - Pasarla a posfija y usar pilas y un arbol de expresion

ARBOL DE EXPRESION



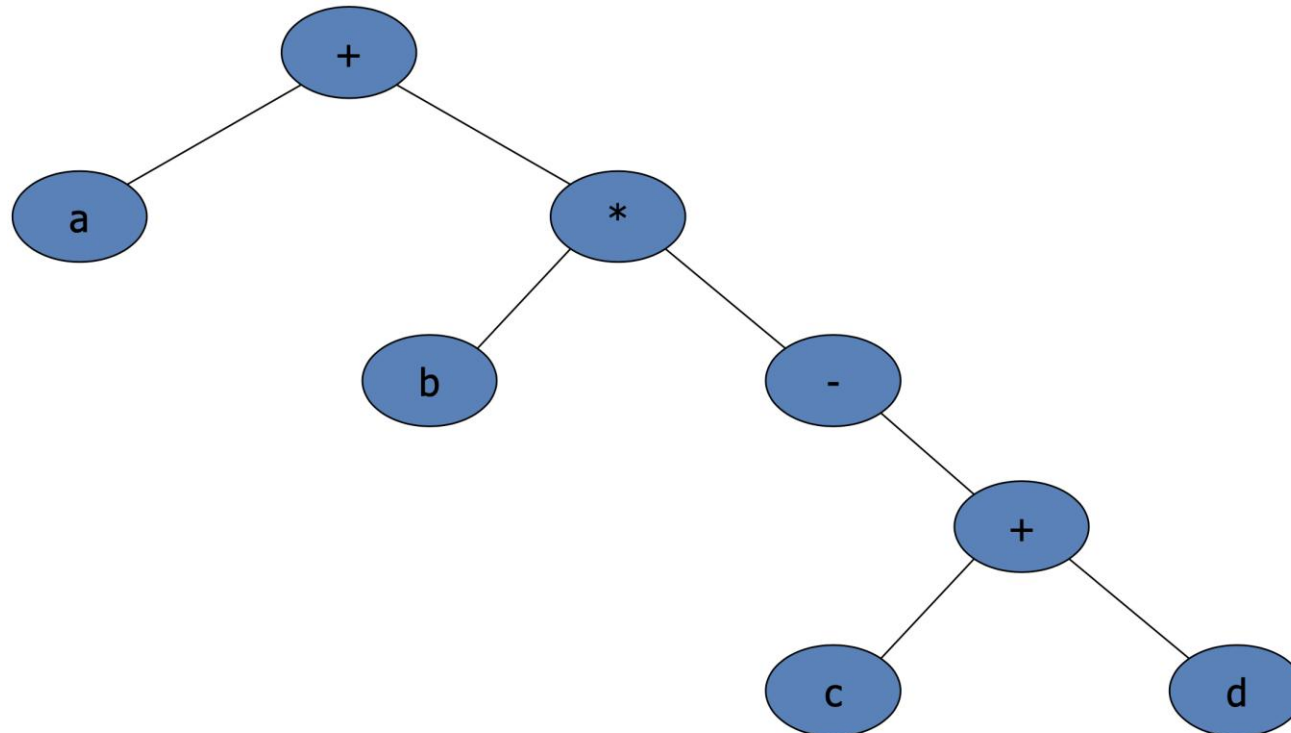
- Árboles que representan expresiones en memoria
 - Todos los operadores tienen dos operandos
 - La raíz puede contener el operador
 - Hijo izq: operando 1, Hijo derecho: operando 2
 - Ejemplo: $(a+b)$



EJERCICIO EN CLASE



- Construya arboles de expresion para:
 - $[X+(Y*Z)] * (A-B)$
- Deducir las expresiones de los siguientes A.B.



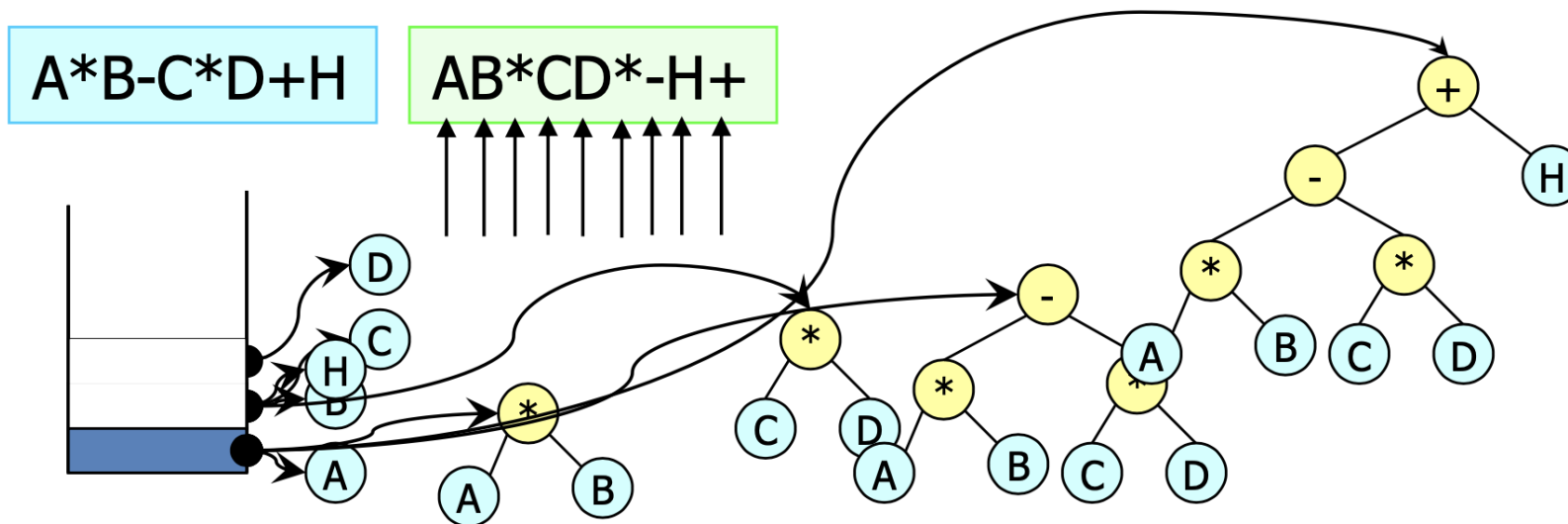
EVALUAR UNA EXPRESION ARITMETICA EN INFIJA

- La expresion se transforma a la expresion posfija
 - Esto, ya sabemos como hacer
- Crear un arbol de expresion
 - Para esto se va a usar una pila y un arbol de caracteres
- Usando el arbol, evaluar la expresion

CREAR UN ARBOL DE EXPRESION



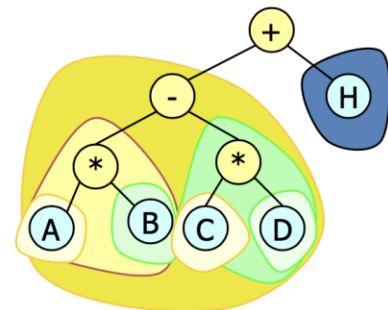
- Los operandos serán siempre nodos hoja del árbol
 - Al revisar un operando, creo una nueva hoja y la recuerdo
- Los operadores serán nodos padre
 - Al revisar un operador, recuerdo las dos últimas hojas creadas y uno todo
 - No debo olvidar el nuevo arbolito que he creado



EVALUACION DE LA EXP. POSTFIJA



- Lo ideal es recuperar los dos operandos, el operador, y ejecutar la opción
 - Que recorrido es el ideal?
 - PostOrden



$(A * B) - (C * D) + H$

Para evaluar el arbol:

Si el arbol tiene un solo nodo y este almacena un operando

El resultado de la evaluacion es el valor de ese operando

Si no

1. Res1 = Evaluo subarbol izquierdo
2. Res2 = Evaluo subarbol derecho
3. Recupero la info de la raiz y efectuo la operación allí indicada, entre Res1 y Res2

ACTIVIDAD EN CLASE



PROBLEMA



Crear un menú con Arboles Binarios que permita generar el registro nuevo de nodo con recorrido de pre-orden, post-orden e InOrden y generar una búsqueda del nodo ingresado.

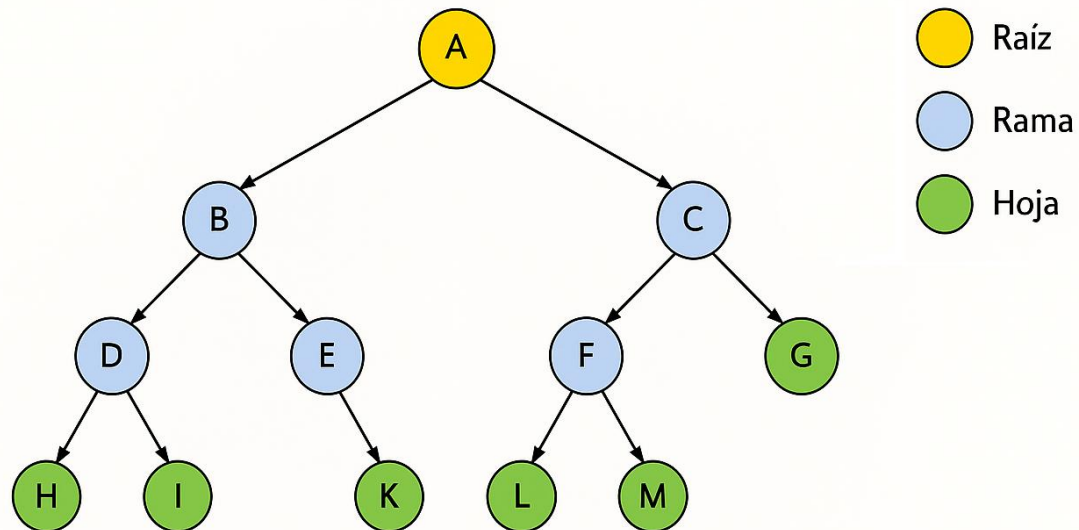
```
Console.WriteLine(" |-----|");
Console.WriteLine(" |           Arboles Binarios           |");
Console.WriteLine(" |-----|-----|");
Console.WriteLine(" | 1. Registrar un nuevo nodo                |");
Console.WriteLine(" | 2. Recorrido en pre-orden                  |");
Console.WriteLine(" | 3. Recorrido en post-orden                 |");
Console.WriteLine(" | 4. Recorrido en In-orden                   |");
Console.WriteLine(" | 5. Buscar nodo                            |");
Console.WriteLine(" |-----|-----|");
Console.Write("  Escoja una opción: ");
```

CONCLUSIONES



Un árbol se puede definir como una estructura jerárquica y en forma no lineal, aplicada sobre una colección de elementos u objetos llamados nodos. (Cairó & Guardati, 2006).

Tipos de nodos en un Árbol



¿PREGUNTAS O COMENTARIOS?



COMPROBACIÓN DEL LOGRO





Cuestionario en la plataforma

Nivel	Rango
Nivel 4	17 – 20
Nivel 3	13 – 16
Nivel 2	9 – 12
Nivel 1	0 - 8

Comprobación del Logro

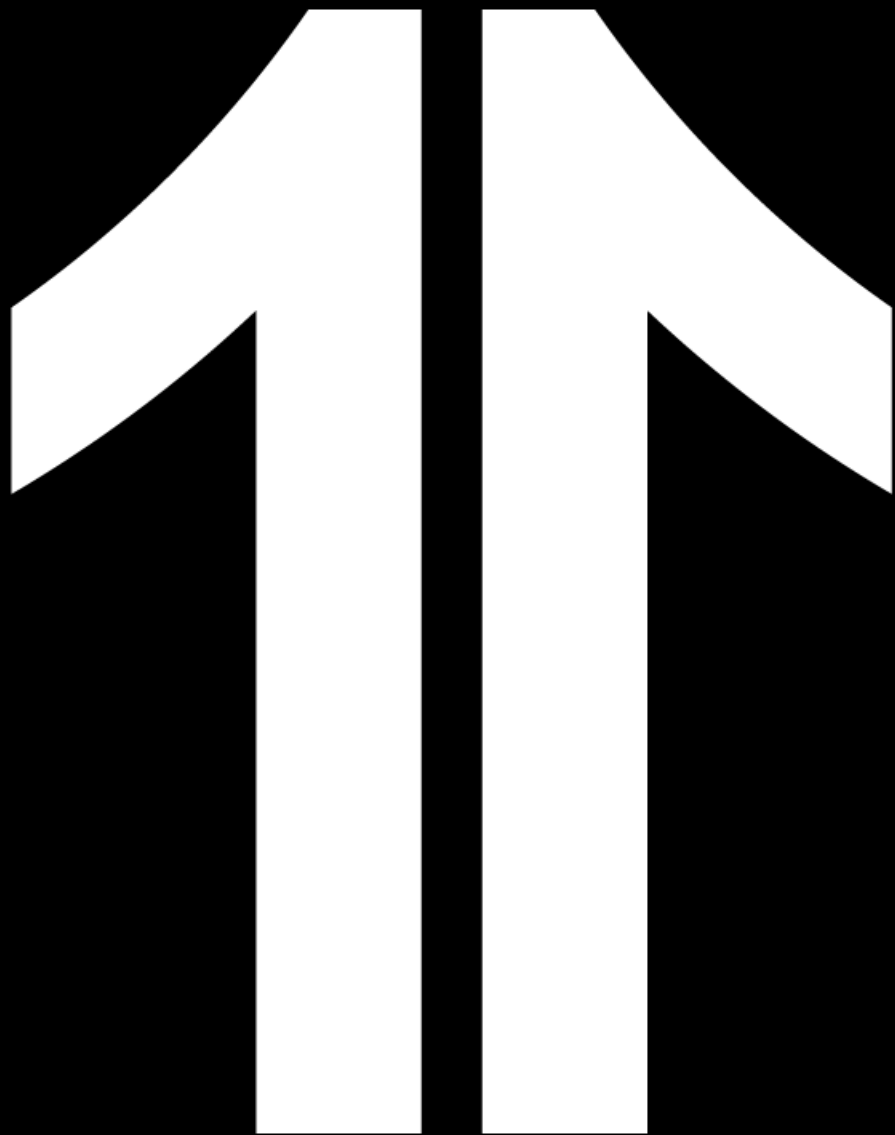


Criterio	Nivel 4 (Excelente)	Nivel 3 (Bueno)	Nivel 2 (Regular)	Nivel 1 (Deficiente)
Comprensión conceptual	Demuestra comprensión completa de los conceptos clave.	Comprende la mayoría de los conceptos clave.	Muestra comprensión parcial, con algunas confusiones.	No demuestra comprensión clara de los conceptos.
Aplicación de conocimientos	Aplica correctamente los conceptos en ejemplos o escenarios.	Aplica conceptos con algunos errores menores.	Aplica de forma superficial o con errores evidentes.	No logra aplicar los conceptos o lo hace incorrectamente.
Claridad y precisión en respuestas	Las respuestas son claras, bien estructuradas y precisas.	Las respuestas son mayormente claras, con leves imprecisiones.	Respuestas poco claras o con ideas mal organizadas.	Respuestas confusas, incompletas o incoherentes.
Cobertura del contenido esperado	Responde completamente las 5 preguntas, desarrollando bien cada una.	Responde correctamente 4 de las 5 preguntas.	Responde correctamente 2 o 3 preguntas.	Responde solo 1 pregunta correctamente o no responde.

BIBLIOGRAFIA REFERENCIAL



- Ceballos Sierra, F. Microsoft C#: Curso de Programación (2a.ed.) 2014
<https://elibronet.eu1.proxy.openathens.net/es/lc/upnorte/titulos/106417>
- Cesar Liza Avila; Estructura de datos con C/C++



GRACIAS

