

# Introducción

Los árboles son estructuras jerárquicas no lineales compuestas por nodos interconectados que representa relaciones entre elementos. Cada nodo puede tener uno o varios nodos hijos, pero solo pueden tener un padre. Estos se usan para organizar datos de forma eficiente, como en el sistema de archivos de una computadora, bases de datos o algoritmos de búsqueda, por ejemplo.

En esta actividad se implementará un árbol binario en Java, realizando operaciones básicas (como insertar, eliminar y buscar), y demostrar diferentes tipos de recorridos. Adicionalmente, gestionar un sistema de empleados usando árboles binarios para mejorar la búsqueda y gestión de información de los empleados.

# Nodo

Texto

El contenido generado por IA puede ser incorrecto.

La clase Nodo, representa la unidad básica de un árbol binario: un contenedor que guarda un valor y referencias a sus nodos hijo izquierdo y derecho. Los siguientes campos permiten que cada Nodo conozca su contenido y cómo enlazarse para formar subárboles:

* Dato: Valor entero que se usa para ordenar y buscar en el árbol.
* Nombre: Información asociada al nodo.
* Izq: Hijo izquierdo; apunto a otro Nodo o es nulo si no existe.
* Dch: Hijo derecho; lo mismo con su hermano zurdo.

Dentro del constructor se asignan los campos con sus estados iniciales:

* Asignación de clave. Dato recibe el entero D, que será la base para comparar en inserciones y búsquedas.
* Asociación Semántica. Nombre recibe Nom para adjuntar información legible.
* Referencias Nulas. Los hijos se inician con valores nulos, indicando que el nodo comienza sin hijos. A medida que se insertan valores en el árbol, otros nodos se conectan a esas referencias.

El Método toString proporciona una representación de texto útil para depurar, visualizar o revisar los logs.

# Árbol

La clase Arbol es la estructura principal que gestiona todos los nodos que forman el árbol binario. Antes de empezar con los métodos, se declarará una variable raíz de tipo Nodo, este será la entrada principal/primer nodo desde el cual se ramifican todos los demás. También se establece el constructor de la clase Arbol, se ejecuta automáticamente cuando creas una nueva instancia del árbol.

## Funciones

### Agregar

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

El método Agregar inserta un nuevo nodo en el árbol binario, manteniendo el orden basado en el valor Dato. Si el árbol está vacío, ese nodo se convierte en raíz. De lo contrario, se recorre el árbol desde la raíz y se coloca el nuevo nodo en la posición correcta:

* A la izquierda si el dato nuevo es menor.
* A la derecha si es mayor o igual.

El código consiste en lo siguiente:

1. Primero se crea un nuevo nomo con el Dato D y el Nombre Nom.
2. Se cerciora que el árbol está vacío: Si no hay raíz, está vacío, por ende, se asigna un nuevo nodo (que será la raíz).
3. Se prepara para recorrer el árbol con aux, que es el nodo que se usa para recorrer el árbol, y padre, que guardará el nodo anterior.
4. Se encontrará un bucle infinito que solo se rompe cuando se inserta el nodo, y se guarda el nodo actual Aux como Padre.
5. Se compara y se decidirá a qué dirección va el nodo:
   1. Si el nuevo dato es menor que el actual, se va a la izquierdo.
   2. Si el nuevo dato es mayor o igual, se va a la derecha.

### Buscar

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

El método Buscar, encuentra un nodo dentro del árbol que tenga el valor de Dato igual a D. Si se encuentra, se devuelve ese nodo, si no, su resultado será nulo.

1. Se usa la variable auxiliar que empieza en la raíz del árbol, este se usará para recorrer el árbol sin modificar la raíz original.
2. Mientras haya nodos por revisar:
   1. Si el dato del nodo actual es igual al que se busca, se devuelve el nodo.
   2. Si el dato buscado es menor que el nodo actual, se mueve a la rama izquierda.
   3. Si el dato es mayor o igual, se mueve a la derecha.
3. Si no se ha encontrado, se devuelve NULL.

### Eliminar

Texto

El contenido generado por IA puede ser incorrecto.

El método de eliminación usa una función recursiva que busca un nodo de valor D y lo elimina del árbol:

1. Si llegamos a un nodo nulo, significa que el valor no existe en el árbol.
2. Si el valor a eliminar es menor que el actual, se busca en la rama izquierda.
3. Si el valor es igual o mayor, estamos en el nodo a eliminar:
   1. Si el nodo solo tiene hijo derecho (o ninguno), se reemplaza por el diestro.
   2. Si el nodo tiene solo uno zurdo, se reemplaza por el zurdo.
   3. Si el nodo tiene los dos, se busca el nodo más pequeño del subárbol derecho, se copia su valor al nodo actual y liego se elimina el sucesor del diestro.

### Encontrar

Texto

El contenido generado por IA puede ser incorrecto.

Este encuentra el nodo más pequeño dentro del subárbol, recibe como parámetro un nodo cualquiera, recorre hacia la izquierda hasta que ya no haya más zurdos y devuelve el nodo más pequeño del subárbol. Usualmente en un ABB, los valores pequeños están por la izquierda, el bucle sigue bajando por la rama izquierda hasta llegar al último nodo.

## Recorridos

Recorrer un árbol significa visitar cada nodo exactamente una vez, siguiendo una estrategia específica. Dependiendo del orden, se obtiene diferentes resultados como:

* Una lista ordenada de valores.
* Una reconstrucción del árbol.
* Una forma de imprimir o procesar datos.

### Inorden

Texto

El contenido generado por IA puede ser incorrecto.

La orden va a lo siguiente: Izquierda 🡪 Raíz 🡪 Derecha. Como el árbol es binario, se devuelve los valores ordenados de menor a mayor.

### Pre-Orden

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Raíz 🡪 Izquierda 🡪 Derecha. Es útil para copiar o serializar un árbol.

### Post-Orden

Texto

El contenido generado por IA puede ser incorrecto.

Izquierda 🡪 Derecha 🡪 Raíz. Este es útil para eliminar nodos o liberar memoria, ya que visita hijos antes que el padre.

# Principal

## Datos

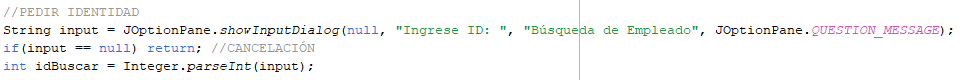


## Población

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

## Pedir ID



## Búsqueda Secuencial

Texto

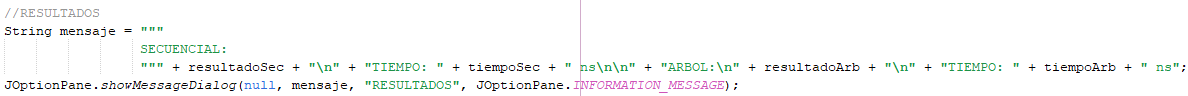
El contenido generado por IA puede ser incorrecto.

## Búsqueda en el árbol

Texto, Carta

El contenido generado por IA puede ser incorrecto.

## Resultados



## Función Secuencial

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.