

ANEXOS: UTILIZA RECURSIVIDAD EN LA PROGRAMACIÓN

SESION 1 / SEMANA 14

Anexo 01 : Tarea Participacion 1401

Fecha Entrega : 03/12/2024 8:00 pm

Nombre de Estudiante : Juan Piero Vincha Loza

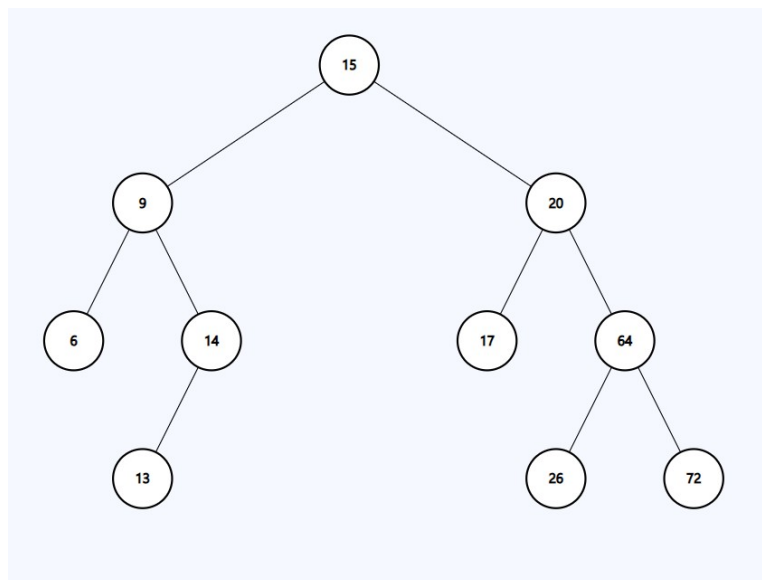
1. Implementar el código JAVA ([428-1401.java](#)), que permite crear un árbol binario.

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  POLYGLOT NOTEBOOK
>> A01 git:(main) 04:20:04
• java Main.java
Ingrese números para el árbol binario (separados por espacio):
15 9 6 14 13 20 17 64 26 72

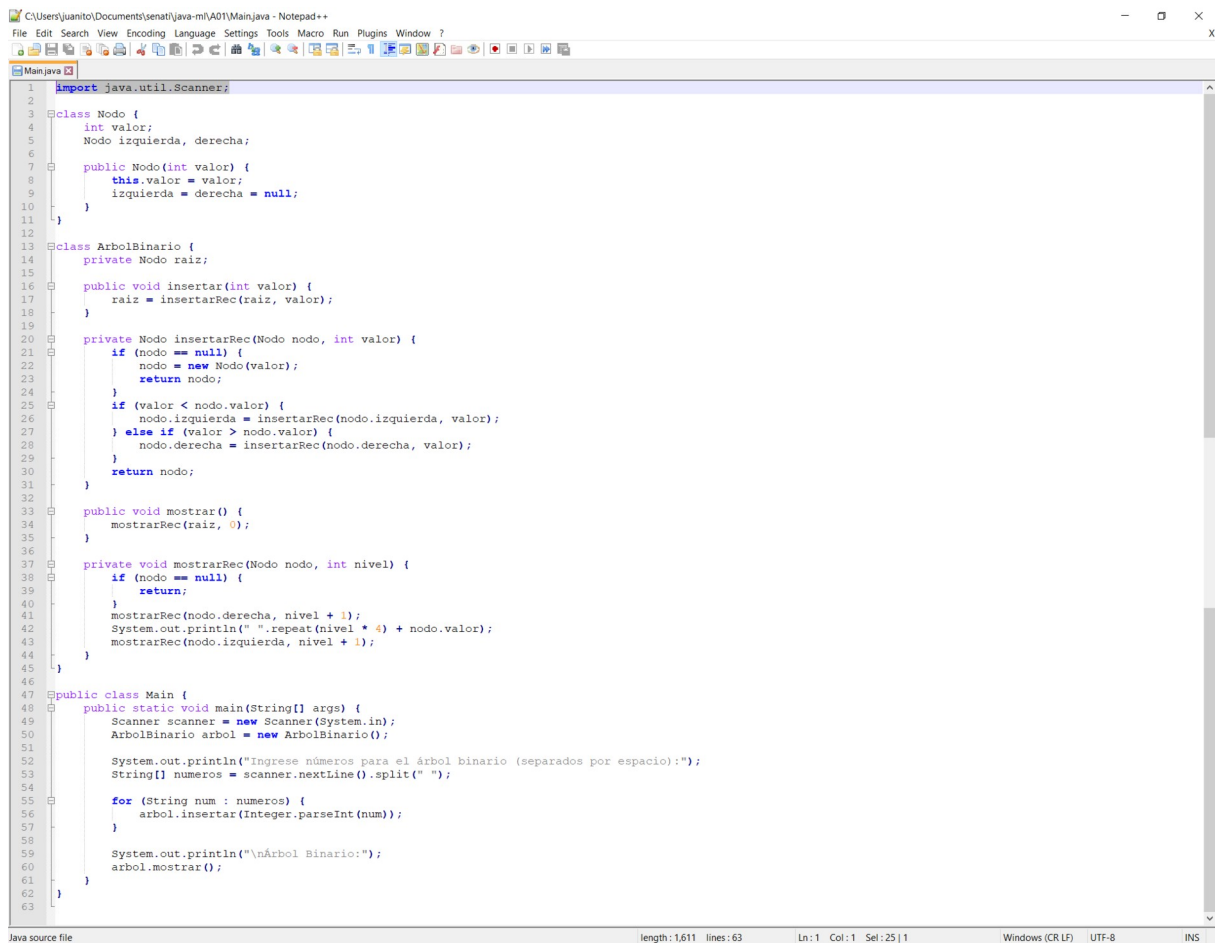
Arbol Binario:
      72
     /  \
    64   26
   /  \
  20   17
 /  \
15   14
 /  \
9    13
 /  \
6    6

>> A01 git:(main) 04:20:42
```

La raíz del árbol es 15



Un árbol binario es una estructura de datos jerárquica donde cada nodo tiene como máximo dos hijos. Es ampliamente utilizado en algoritmos de búsqueda, compiladores y estructuras de bases de datos.



```

1  import java.util.Scanner;
2
3  class Nodo {
4      int valor;
5      Nodo izquierda, derecha;
6
7      public Nodo(int valor) {
8          this.valor = valor;
9          izquierda = derecha = null;
10     }
11 }
12
13 class ArbolBinario {
14     private Nodo raiz;
15
16     public void insertar(int valor) {
17         raiz = insertarRec(raiz, valor);
18     }
19
20     private Nodo insertarRec(Nodo nodo, int valor) {
21         if (nodo == null) {
22             nodo = new Nodo(valor);
23             return nodo;
24         }
25         if (valor < nodo.valor) {
26             nodo.izquierda = insertarRec(nodo.izquierda, valor);
27         } else if (valor > nodo.valor) {
28             nodo.derecha = insertarRec(nodo.derecha, valor);
29         }
30         return nodo;
31     }
32
33     public void mostrar() {
34         mostrarRec(raiz, 0);
35     }
36
37     private void mostrarRec(Nodo nodo, int nivel) {
38         if (nodo == null) {
39             return;
40         }
41         mostrarRec(nodo.derecha, nivel + 1);
42         System.out.println(" ".repeat(nivel * 4) + nodo.valor);
43         mostrarRec(nodo.izquierda, nivel + 1);
44     }
45 }
46
47 public class Main {
48     public static void main(String[] args) {
49         Scanner scanner = new Scanner(System.in);
50         ArbolBinario arbol = new ArbolBinario();
51
52         System.out.println("Ingrese números para el árbol binario (separados por espacio):");
53         String[] numeros = scanner.nextLine().split(" ");
54
55         for (String num : numeros) {
56             arbol.insertar(Integer.parseInt(num));
57         }
58
59         System.out.println("\nÁrbol Binario:");
60         arbol.mostrar();
61     }
62 }
63

```

Explicación del código:

a) Clase Nodo: Define los nodos del árbol, que almacenan un valor y referencias a los nodos izquierdo y derecho.

b) Clase ArbolBinario: Implementa la lógica para insertar valores y mostrar el árbol.

- insertar: Añade un valor al árbol.
- mostrar: Visualiza el árbol en formato jerárquico.

c) Clase Main: Punto de entrada del programa donde el usuario puede interactuar con el árbol

2.-Implementar el factorial de un numero

```

12 //Ejemplo para funcion factorial
13 int factorial(int n)
14 {
15     if (n==0) //Caso base
16     {
17         n=1;
18     }else //Caso general
19     {
20         n=*factorial(n-1);
21     }
22     return n;
23 }

```

Figura N° 10: Recursividad, Caso Factorial en Java.

```

Main.java M X
A01 > BinaryTree > Main.java
50 public class Main {
51
52     // Método para calcular el factorial de un número
53     public static int factorial(int n) {
54         if (n == 0) { // Caso base
55             return 1;
56         } else { // Caso general
57             return n * factorial(n - 1);
58         }
59     }
60 }

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS POLYGLOT NOTEBOOK

```

>> BinaryTree git:(main)
● java Main.java
Ingrese números para el árbol binario (separados por espacio):
10 5 15 3 7 12 18

Árbol Binario:
      18
     /  \
    15   12
   /  \  /  \
  10  7 5   3
 /  \
5   3

Ingrese un número para calcular su factorial:
5
El factorial de 5 es: 120
>> BinaryTree git:(main)

```

Cambios agregados a Main.java y resultado

```

76 // Calcular el factorial de un número
77 System.out.println("\nIngrese un número para calcular su factorial:");
78 int numero = scanner.nextInt();
79 System.out.println("El factorial de " + numero + " es: " + factorial(numero));

```

Después de mostrar el árbol, el programa pide al usuario un número y calcula su factorial usando el método factorial.