

1 UTILIZANDO ESTRUTURA DE DADOS ÁRVORE BINÁRIA

1.1 Código fonte da resolução

```
public class No
{
    private long id;
    private No esq, dir;

    public No(long id, No esq, No dir)
    {
        this.id = id;
        this.esq = esq;
        this.dir = dir;
    }

    public long getId()
    {
        return this.id;
    }

    public No getEsq()
    {
        return this.esq;
    }

    public No getDir()
    {
        return this.dir;
    }

    public void setEsq(No esq)
    {
        this.esq = esq;
    }

    public void setDir(No dir)
    {
        this.dir = dir;
    }
}

public class ArvoreBinaria {
    private No raiz;

    public ArvoreBinaria()
    {
        this.raiz = null;
    }
}
```

```

public void inserir(long id)
{
    raiz = inserirNo(raiz, id);
}

private No inserirNo(No raiz, long id)
{
    if (raiz == null) {
        return new No(id, null, null);
    }

    if (id < raiz.getId()) {
        raiz.setEsq(inserirNo(raiz.getEsq(), id));
    } else if (id > raiz.getId()) {
        raiz.setDir(inserirNo(raiz.getDir(), id));
    }

    return raiz;
}

public long esquerdaFestiva() {
    return esquerdaFestivaRecursivo(raiz);
}

private long esquerdaFestivaRecursivo(No no) {
    if (no == null)
        return 0;

    int contador = 0;
    if (no.getEsq() != null)
        contador = 1;

    contador += esquerdaFestivaRecursivo(no.getEsq());
    contador += esquerdaFestivaRecursivo(no.getDir());

    return contador;
}

public void imprimirEstruturaComRecuo() {
    System.out.println("Árvore Binária:");
    imprimirEstruturaRecursivo(raiz, 0);
}

private void imprimirEstruturaRecursivo(No no, int nivel) {
    for (int i = 0; i < nivel; i++) {
        System.out.print(" ");
    }
    if (no != null) {
        System.out.println(no.getId());
    } else {

```

```

        System.out.println("-");
        return;
    }

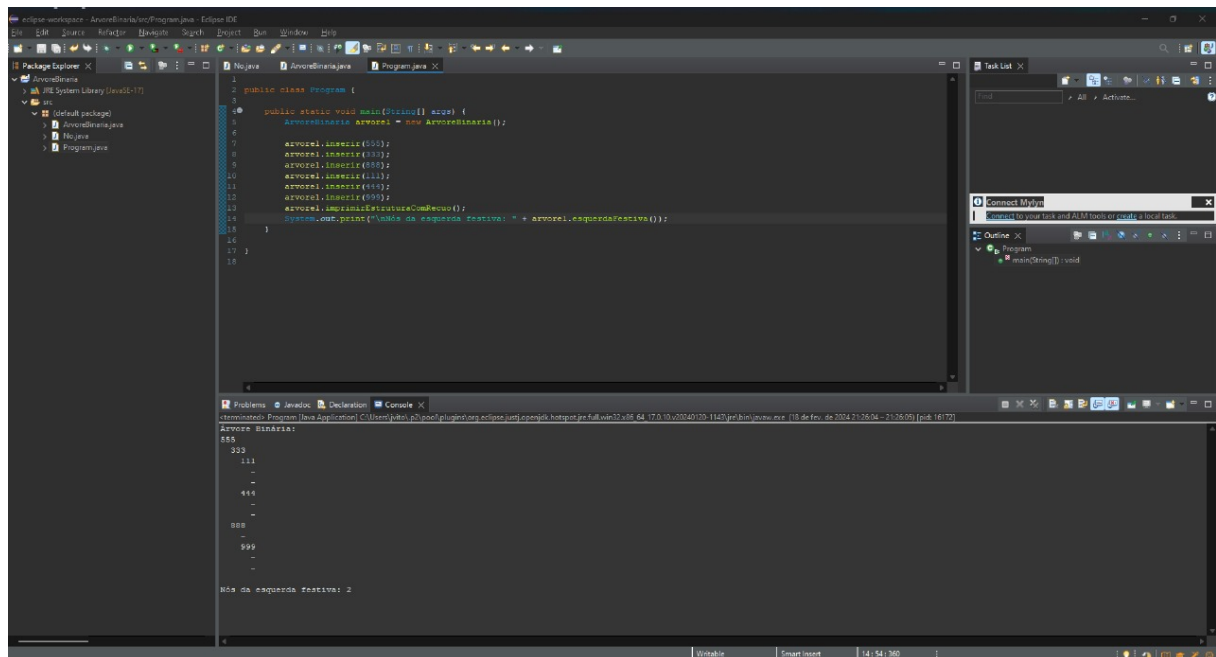
    imprimirEstruturaRecursivo(no.getEsq(), nivel + 1);
    imprimirEstruturaRecursivo(no.getDir(), nivel + 1);
}
}
public class Program {

    public static void main(String[] args) {
        ArvoreBinaria arvore1 = new ArvoreBinaria();

        arvore1.inserir(555);
        arvore1.inserir(333);
        arvore1.inserir(888);
        arvore1.inserir(111);
        arvore1.inserir(444);
        arvore1.inserir(999);
        arvore1.imprimirEstruturaComRecuo();
        System.out.print("\nNós da esquerda festiva: " + arvore1.esquerdaFestiva());
    }
}

```

1.2 Programa rodando



1.3 Repositório do projeto no GitHub

<https://github.com/jvitor-alol/Arvore-Binaria>