

```

package com.metodosnumericos.view;

import net.objecthunter.exp4j.Expression;
import net.objecthunter.exp4j.ExpressionBuilder;

/**
 *
 * @author angel
 */

public class Main {
    public static void main(String[] args) {
        // Expresión algebraica ingresada como cadena
        String algebraicExpression = "2 * x ^ 2 + 3";

        // Crear una instancia de ExpressionBuilder de exp4j
        Expression expression = new ExpressionBuilder(algebraicExpression)
            .variable("x") // Definir la variable 'x'
            .build();

        // Definir el valor de 'x'
        double x = 0.0; // Por ejemplo, x = 5.0

        // Asignar el valor a la variable 'x'
        expression.setVariable("x", x);

        // Evaluar la expresión
        double result = expression.evaluate();

        // Imprimir el resultado
        System.out.println("Resultado: " + result);

        double root = solveBisection(expression, 0, 10);

        System.out.println("Biseccion con un rango entre 0 y 10: " + root);
    }

    public static double solveBisection(Expression expression, double x0, double
x1) {
        // Definir la tolerancia
        double tolerancia = 0.0001;
        double xm = 0;
    }

```

```

// Bucle de biseccion
while (Math.abs(x1 - x0) > tolerancia) {
    // Calcular el punto medio
    xm = (x0 + x1) / 2;

    expression.setVariable("x", xm);

    // Evaluar la función en el punto medio
    double fm = expression.evaluate();

    expression.setVariable("x", x0);

    // Determinar el siguiente intervalo
    if (fm * expression.evaluate() < 0) {
        x1 = xm;
    } else {
        x0 = xm;
    }
}

// Devolver la raíz
return xm;
}
}

```