# ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

## Facultad de Informática y Electrónica

## Validación y Verificación de Software


**Integrantes:** - Homero Barragán (6766)

- Tatiana Carrillo (6768)

- Emily Domínguez (6771)

- Steeven Romero (6804)

**Curso:** Octavo A

**Fecha:** 2022 – 10 – 03

### Cálculo de las raíces de un polinomio de segundo grado

## Análisis

### Requisitos funcionales

| | |
|---|---|
| **Identificación del requerimiento:** | RF01 |
| **Nombre del requerimiento:** | Verificar datos |
| **Características:** | El sistema verificará que los datos sean correctos. |
| **Descripción del requerimiento:** | El sistema verificará que los datos ingresados sean numéricos y que el valor de a sea diferente de 0. |

| | |
|---|---|
| **Identificación del requerimiento:** | RF02 |
| **Nombre del requerimiento:** | Calcular las raíces |
| **Características:** | El sistema calculará las raíces de un polinomio de segundo grado. |
| **Descripción del requerimiento:** | El sistema calculará las raíces reales e imaginarias conjugadas de un polinomio de segundo grado. |

## Implementación

### Polinomio.java

```
package espoch.edu.ec.raicespolinomio;

public class Polinomio {

    public double a;
    public double b;
    public double c;
    public double delta;
    public double x1;
```

```
    public double x2;

    public Polinomio(double a, double b, double c) throws Exception {

        if (a == 0)
            throw new Exception("No es un polinomio de grado 2");

        this.a = a;
        this.b = b;
        this.c = c;
        this.delta = (double) Math.pow(this.b,2)-(4*this.a*this.c);
    }

    public void CalcularRaiz(){
        if(this.delta >= 0){
            this.x1 = (double) (-b + Math.sqrt(this.delta))/(2*a);
            this.x2 = (double) (-b - Math.sqrt(this.delta))/(2*a);
            System.out.printf("x1 = %.2f ", x1);
            System.out.printf("\nx2 = %.2f ", x2);
        }
        else {
            double real = -b / (2*a);
            double imaginary = Math.sqrt(-this.delta) / (2*a);
            System.out.printf("x1 = %.2f+%.2fi",
                              real, imaginary);
            System.out.printf("\nx2 = %.2f-%.2fi",
                              real, imaginary);
        }
    }

}
```

**RaicesPolinomio.java**

```
public class RaicesPolinomio {

    static double a;
    static double b;
    static double c;

    public static void Ingreso()throws Exception{
        Scanner leer = new Scanner (System.in);
        try{
            System.out.print("Ingrese el valor de a: ");
            a = leer.nextDouble();
            System.out.print("Ingrese el valor de b: ");
            b = leer.nextDouble();
            System.out.print("Ingrese el valor de c: ");
            c = leer.nextDouble();
        }catch(Exception e) {
            throw new Exception("Ingrese sólo números");
        }
    }

    public static void main(String[] args) {
        try {
            Ingreso();
            Polinomio miPolinomio = new Polinomio(a, b, c);
```

```
            miPolinomio.CalcularRaiz();
        } catch (Exception e) {
            System.out.println(e);
        }
    }

}
```

## Pruebas unitarias

| Requisito | Función | Caso de prueba | Datos de entrada | Resultado esperado | Datos de salida | Evaluación de la ejecución |
|---|---|---|---|---|---|---|
| RF01. Verificar datos | Ingreso | CP1 | (1, 2, h) | Ingrese sólo números | Ingrese sólo números | Prueba aprobada |
| | | CP2 | (3, g, 8) | Ingrese sólo números | Ingrese sólo números | Prueba aprobada |
| | | CP3 | (a, b, c) | Ingrese sólo números | Ingrese sólo números | Prueba aprobada |
| | | CP4 | (0, 1, 2) | No es un polinomio de grado 2 | No es un polinomio de grado 2 | Prueba aprobada |
| | | CP5 | (g, 2, -3) | Ingrese sólo números | Ingrese sólo números | Prueba aprobada |
| | | CP6 | (1, b, -3) | Ingrese sólo números | Ingrese sólo números | Prueba aprobada |
| | | CP7 | (a, 9, j) | Ingrese sólo números | Ingrese sólo números | Prueba aprobada |
| | | CP8 | (e, u, 7) | Ingrese sólo números | Ingrese sólo números | Prueba aprobada |
| RF02. Calcular las raíces | CalcularRaiz | CP9 | (1, 0, 9) | Raíces imaginarias | x1 = -0,00+3,00i x2 = -0,00-3,00i | Prueba aprobada |
| | | CP10 | (1, 2, -3) | Raíces reales | x1 = 1,00 x2 = -3,00 | Prueba aprobada |
| | | CP11 | (4, 100000, -6) | Ingrese valores de 5 cifras | Ingrese valores de 5 cifras | Prueba aprobada |
| | | CP12 | (100000, 7, -8) | Ingrese valores de 5 cifras | Ingrese valores de 5 cifras | Prueba aprobada |
| | | CP13 | (-9, 13, 100000) | Ingrese valores de 5 cifras | Ingrese valores de 5 cifras | Prueba aprobada |
| | | CP14 | (20, 13, 15) | Raíces imaginarias | x1 = -0,33+0,80i x2 = -0,33-0,80i | Prueba aprobada |
| | | CP15 | (-4, -7, -5) | Raíces imaginarias | x1 = -0,88-0,70i x2 = -0,88+0,70i | Prueba aprobada |

### CP1

```
Ingrese el valor de a: 1
Ingrese el valor de b: 2
Ingrese el valor de c: h
java.lang.Exception: Ingrese solo numeros
------------------------------------------------------------------------
BUILD SUCCESS
```

### CP2

```
Ingrese el valor de a: 0
Ingrese el valor de b: 1
Ingrese el valor de c: 2
java.lang.Exception: No es un polinomio de grado 2
------------------------------------------------------------------------
BUILD SUCCESS
```

### CP3

```
Ingrese el valor de a: g
java.lang.Exception: Ingrese solo numeros
------------------------------------------------------------------------
BUILD SUCCESS
```

### CP4

```
Ingrese el valor de a: 1
Ingrese el valor de b: b
java.lang.Exception: Ingrese solo numeros
------------------------------------------------
BUILD SUCCESS
------------------------------------------------
```

**CP5**

```
Ingrese el valor de a: 1
Ingrese el valor de b: 0
Ingrese el valor de c: 9
x1 = -0,00+3,00i
x2 = -0,00-3,00i
------------------------------------
BUILD SUCCESS
------------------------------------
```

**CP6**

```
Ingrese el valor de a: 1
Ingrese el valor de b: 2
Ingrese el valor de c: -3
x1 = 1,00
x2 = -3,00
------------------------------------
BUILD SUCCESS
------------------------------------
```

**CP7**

```
Ingrese el valor de a: 4
Ingrese el valor de b: 100000
Ingrese el valor de c: -6
java.lang.Exception: Ingrese valores de 5 cifras
------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------
```

**CP8**

```
Ingrese el valor de a: 100000
Ingrese el valor de b: 7
Ingrese el valor de c: -8
java.lang.Exception: Ingrese valores de 5 cifras
------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------
```

**CP9**

```
Ingrese el valor de a: -9
Ingrese el valor de b: 13
Ingrese el valor de c: 100000
java.lang.Exception: Ingrese valores de 5 cifras
------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------
```