

Práctica 0

Ecuaciones cuadráticas

Pepper Pots (A01166611) y Anthony Stark (A01160611)

10 de agosto, 2016.

Tabla de contenido

1. Introducción	1
2. Desarrollo	1
3. Pruebas.....	3
4. Conclusión	3
5. Agradecimientos	4
6. Referencias	4

Este reporte de laboratorio fue elaborado para el curso Tc2026 *Desarrollo de aplicaciones web* en el Tecnológico de Monterrey, Campus Estado de México.

1. Introducción

Las raíces de una ecuación cuadrática $ax^2 + bx + c = 0$ pueden ser calculadas mediante la siguientes fórmula:

$$x = (-b \pm \text{sqrt}(b^2 - 4ac)) \div 2a$$



NOTA INTERESANTE

A la fórmula para determinar la solución a una ecuación cuadrática se le llama *fórmula general*, pero en México se le conoce también con el nombre folclórico del “chicharronero”. Este es debido a que se supone que todo mundo la conoce, y ¡hasta el señor que vende chicharrones la sabe utilizar!

En este reporte se muestra la manera de determinar el número de raíces con las que cuenta una ecuación cuadrática utilizando el lenguaje de programación JavaScript [\[MDN\]](#). La sección 2 consiste del desarrollo y el código de la solución, la sección 3 describe la manera en que se probó el programa, y finalmente incluimos nuestras conclusiones.

2. Desarrollo

Se diseñó una función llamada `numRealRoots()` para calcular el número de raíces con las que cuenta una ecuación cuadrática. De acuerdo con [\[Baldor\]](#), esto se puede determinar calculando el discriminante de la ecuación y luego probando su valor.

El discriminante es la expresión que va dentro de la raíz cuadrada de la fórmula general:

$$b^2 - 4ac$$

Hay tres posibilidades:

- Si el discriminante es negativo, el número de raíces reales es cero.
- Si el discriminante es igual a cero, el número de raíces reales es uno.
- Si el discriminante es positivo mayor a cero, el número de raíces reales es dos.

En nuestro código se definieron adicionalmente dos funciones auxiliares: `discriminant()` y `readNumber()`, con el fin de hacer más legible nuestro programa.

El código completo se muestra a continuación:

```
/*-----  
 * Práctica 0: Ecuaciones cuadráticas  
 * Fecha: 10-Agosto-2016  
 * Autores:  
 *     A01166611 Pepper Pots  
 *     A01160611 Anthony Stark  
 *-----*/  
  
'use strict';  
  
function numRealRoots(a, b, c) {  
  let d = discriminant(a, b, c);  
  if (d < 0) {  
    return 0;  
  } else if (d === 0) {  
    return 1;  
  } else {  
    return 2;  
  }  
}  
  
function discriminant(a, b, c) {  
  return b * b - 4 * a * c;  
}  
  
// Leer un número del teclado (entrada estándar).  
function readNumber(prompt, cb) {  
  process.stdout.write(prompt);  
  process.stdin.once('data', input => {  
    let num = parseFloat(input);  
    if (isNaN(num)) {  
      readNumber(prompt, cb);  
    } else {  
      cb(num);  
    }  
  });  
}  
  
// Obtener los valores de A, B y C del teclado y calcular  
// e imprimir el número de raíces reales.  
readNumber('A = ', a =>  
  readNumber('B = ', b =>  
    readNumber('C = ', c => {  
      process.stdin.pause();  
      console.log('Number of real roots:', numRealRoots(a, b, c));  
    })  
  )  
);
```

3. Pruebas

El programa para calcular el número de raíces reales fue probado usando los valores de la siguiente tabla que considera los tres posibles casos:

A	B	C	Resultado
1	2	3	0
1	10	2	2
2	16	32	1

La siguiente imagen muestra la salida de tres ejecuciones del programa produciendo los resultados esperados:

```
arielortiz:~/workspace/practica0 $ node cuadratica.js
A = 1
B = 2
C = 3
Number of real roots: 0
arielortiz:~/workspace/practica0 $ node cuadratica.js
A = 1
B = 10
C = 2
Number of real roots: 2
arielortiz:~/workspace/practica0 $ node cuadratica.js
A = 2
B = 16
C = 32
Number of real roots: 1
arielortiz:~/workspace/practica0 $
```

4. Conclusión

Una percepción común es que un programa solo puede ser “útil” si es extenso. El determinar el número de raíces reales de una ecuación cuadrática es un ejemplo de un programa que es suficientemente complicado como para que valga la pena ser automatizado, pero que a su vez requiere de relativamente pocas instrucciones. Tal como se demostró en esta práctica de laboratorio, es bastante fácil escribir un programa que requiere de cálculos matemáticos rutinarios.

La sección del programa que resultó un poco más complicada fue la correspondiente a la lectura de números desde el teclado. Nos resultó confuso el mecanismo de llamadas asíncronas que usa Node al momento de leer el *stream* de la entrada estándar. Este esquema es muy distinto a la manera como funcionan otros lenguajes de programación tales como Python o C. Sin embargo, finalmente

entendimos cómo funciona y pudimos terminar exitosamente nuestra práctica.

5. Agradecimientos

Queremos dar las gracias a nuestra amiga y colega Natasha Romanova por sugerirnos consultar el libro de [Baldor] así como por su seductora sonrisa.

6. Referencias

- [Baldor] J. Aureliano Baldor. *Álgebra*. Grupo Patria Cultural. México, 2007. p. 265.
- [MDN] Mozilla Developer Network. *JavaScript Reference*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference> (Consultada el 10 de agosto de 2016).