**Informe Requerimientos Funcionales.**

**Problema:** Encontrar las raíces de cualquier polinomio hasta de grado 10.

**Requerimientos Funcionales.**

**El sistema debe permitir:**

(n=exponente no negativo)

**R1**: Ingresar un polinomio(Números reales) que desea resolver con grado n<=10.

**R2**: Generar un polinomio con números Reales aleatoriamente hasta n<=10.

**R3**: Mostrar todas las raíces del polinomio ya sea entregado como una entrada o generado aleatoriamente .En caso de que las soluciones del polinomio sean complejas se mostrará un mensaje diciendo que el polinomio en un número complejo y las raíces (si las tiene);

**R4**: Mostrar el método que se utilizó para encontrar las raíces del polinomio dado.

**Requerimientos No Funcionales.**

Rendimiento: En el momento que el polinomio tenga un n muy grande o tenga una gran cantidad de términos, el programa tendrá problemas para procesar el algoritmo dependiendo de la capacidad del equipo

Eficiencia: El polinomio se podrá resolverse completamente dependiendo:

La cantidad de elementos y de su exponente ya que un polinomio demasiado grande tardará mucho en resolverse.

En caso de que las raíces de un polinomio sean complejas el programa no lo resolverá sino que se mostrará un mensaje informativo

Usabilidad: el usuario solo contará en la interfaz los espacios correspondientes para escribir los coeficientes del polinomio en el cual deberá escribirlo de manera lógica para que el programa lo entienda y funcione correctamente de lo contrario cualquier tipo de error al escribir el polinomio no se podrá hacer las operaciones para hallar las raíces

**Recopilacion de informacion**:

Para resolver el problema de cómo hallar raíces de polinomios hasta grado 10 se tuvieron en cuenta varios aspectos teóricos matemáticos acerca del significado de un polinomio, las clases o tipos de polinomios que existen ,grado de polinomio y cómo se solucionan cada uno de ellos de carácter matemático, todo ello con el fin de entender tales conceptos y acercarse mejor al problema para poder encontrar la mejor solución.

***Polinomio:***

Un polinomio en la variable x es una expresión algebraica formada solamente por la suma de términos de la forma **ax^n,** donde **a** es cualquier número y **n** es el número entero no negativo.

Componentes:

**Término** :Un término es una parte de una expresión algebraica. Los términos se separan entre sí por los signos de suma (+) o resta (-).

**Coeficiente numérico**: es el factor numérico que acompaña la variable.

**Término constante**: es el coeficiente numérico que no contiene variable.

***Grado del polinomio:***

Si el polinomio es en una variable, el **grado** del polinomio está determinado por el término que contiene el mayor exponente.

Si tiene más de una variable, se suman los exponentes de cada término y la suma más alta determina el grado del polinomio.

Fuente:

<https://www.suagm.edu/une/pdf/ciencias/POLINOMIOS.pdf>

***Raíces de un polinomio:***

Se dice que un valor x = a es raíz de un polinomio P(x), cuando al sustituir dicho valor en el polinomio, el resultado es 0; es decir, cuando P(a) = 0.Las raíces de un polinomio, también se llaman ceros del polinomio.

<http://mestreacasa.gva.es/c/document_library/get_file?folderId=500012797810&name=DLFE-711165.pdf>

Las raíces que puede tener un polinomio son de tres tipos: raíces positivas raíces negativas raíces complejas Las raíces también se pueden presentar con valores repetidos

<https://www.uv.mx/personal/aherrera/files/2014/08/05c.-RAICES-DE-UN-POLINOMIO-GENERALIDADES.pdf>

Después de entender los conceptos matemáticos los cuales son fundamentales y son estrictamente relacionados con nuestro problema de interés se hizo la búsqueda de información de métodos que nos pudieran dar la mejores soluciones en relación a eficiencia de cómo encontrar las raíces.

Los métodos matemáticos encontrados para hallar las raíces de un polinomio fueron los siguientes:

* *Método de Newton para polinomios .*
* *Método de Horner*
* *Método de Bairstow.*
* *Método de Ruffini.*
* *Función Cuadrática*

Descripción matemática de los métodos encontrados:

|  |  |  |  |
| --- | --- | --- | --- |
| **Método:** | **Descripción** | **Ventaja** | **Desventaja** |
| ***Método de Horner.*** | El algoritmo de Horner se usa a menudo para convertir entre distintos sistemas numéricos posicionales — en cuyo caso x es la base del sistema numérico, y los coeficientes ai son los dígitos de la representación del número dado en la base x — y puede usarse también si x es una matriz, en cuyo caso la carga computacional se reduce aún más. | La evaluación usando la forma monomial del polinomio de grado-n requiere al menos n sumas y **(n2+n)/2** multiplicación |  |
| ***Método de Newton para polinomios .*** | Es un algoritmo para encontrar aproximaciones de los ceros o raíces de una función real. También puede ser usado para encontrar el máximo o mínimo de una función, encontrando los ceros de su primera derivada | es eficiente en la solución de sistemas de ecuaciones no lineales, converge muy rápidamente y proporciona una muy buena precisión en los resultados | Aunque es muy eficiente, hay situaciones en que presenta dificultades:  En caso especial es las raíces múltiples  En algunos casos es posible que para raíces simples se presenten dificultades por su lenta convergencia. |
| **Método de Bairstow.** | En análisis numérico, el método de Bairstow es un algoritmo eficiente de búsqueda de las raíces de un polinomio real de grado arbitrario. Es un método iterativo, basado en el método de Müller y de Newton Raphson | Es uno de los más completos debido a su capacidad de no solo obtener raíces reales, sino también las imaginarias | El algoritmo de Bairstow tiene orden de convergencia cuadrático como el método de Newton, excepto en el caso de que el polinomio tenga factores cuadráticos de multiplicidad superior a 1, pudiendo ser el orden de convergencia m*e*nor*.* |
| **Ecuacion cuadratica** | Una función cuadrática de una variable es una función polinómica definida por: f(x) = ax x , con  2 + b + c a =/ 0.  También se le llama Trinomio Cuadrático. | Encuentra las raíces de un polinomio de grado 2.  Las raíces de una función cuadrática son los valores de x para los cuales ax x , y son denotadas 2 + b + c = 0.habitualmente como x y 1 x . | Solo se puede utilizar este método para polinomios menores o iguales al grado dos. |

*fuente:*

[*https://es.wikipedia.org/wiki/Algoritmo\_de\_Horner*](https://es.wikipedia.org/wiki/Algoritmo_de_Horner)

<https://www.clubensayos.com/Temas-Variados/Regla-de-RuRegla-de-Ruffini/2642766.html>

<https://es.wikipedia.org/wiki/Regla_de_Ruffini>

<https://es.wikipedia.org/wiki/M%C3%A9todo_de_Bairstow>

<https://es.wikipedia.org/wiki/Funci%C3%B3n_cuadr%C3%A1tica>

**Búsqueda de ideas y soluciones creativas:**

Soluciones creativas para la interfaz.

* Crear una interfaz donde se visualice por separado cada coeficiente junto a todas las variables con exponentes de 0 a 10 ya fijas para que el usuario no cometa errores de escritura a la hora de escribir todo el polinomio.
* Crear una interfaz donde el usuario primero elija el exponente más alto para mostrar las variables necesarias y los espacios para que escriba el coeficiente correspondiente
* Crear una interfaz donde el usuario pueda escribir todo el algoritmo en el que podrá visualizar todas las operaciones básicas que pueda necesitar.

Soluciones para plantear el polinomio en java:

* Separar cada parte del polinomio Ax^n :A (El número que acompaña la variable), x(La variable), n(el exponente )
* Crear un arreglo de números aleatorios Para A,X de 0 a 9 .

**Transición de las ideas a los diseños preliminares:**

Transición de las ideas de la interfaz gráfica:

Se escogió la idea de crear una interfaz donde el usuario primero elija el exponente más alto para mostrar las variables necesarias y los espacios para que escriba el coeficiente correspondiente ya que a vista de nosotros es más eficiente y más organizada que las demás ideas

De los métodos matemáticos sólo se pudieron encontrar tres de ellos, Método de Ruffini, método de Bairstow y la ecuación cuadrática:

* El método de Ruffini encontrado no era muy eficiente ya que no daba un resultado correcto en cuanto a las raíces de los polinomios evaluados y encontrado, por lo tanto esta opción fue descartada.
* En cuanto al método de bairstow, se escogió ya que funciona correctamente con los polinomios evaluados, además de ellos , este método es iterativo, basado en el método de Müller y de Newton Raphson.
* La función cuadrática se escogió ya que las raíces reales encontradas y evaluadas con polinomios de grado 2 fue exacta.

Por lo tanto, se escogió el método de Bairstow y la función cuadrática para solucionar el problema de encontrar las raíces de un polinomio de grado menor o igual a 10.

Además concordaba con la segunda idea creativas propuesta para el diseño de la interfaz y el manejo de cómo se iba a resolver el problema, ya que solamente era necesario ingresar cada uno de los coeficientes

**6.Preparación informe y especificaciones**

Para encontrar las raíces de polinomios hasta grado 10 se dividió el problema en dos métodos, El primero para encontrar las raíces con polinomio de grado menor a 3 y Un problema que resuelve un polinomio de grado máximo 10.

**Especificación del problema:**

**Problema:** Raíces de un polinomio de grado 2

**Entradas:** conocer los coeficientes de la función f(x) = ax^2 + bx +c , los cuales son los valores de las constantes a, b y c.

**Salida:** r1 y r2, los cuales al evaluarlas en la función f(x) = ax^2 + bx +c =0

*Consideraciones:*

las raíces complejas no serán consideradas como soluciones a nuestro problema, por lo tanto solo se le notificará al usuario.

**Pseudocódigo del método: Diagrama de flujo:**

|  |  |
| --- | --- |
|  |  |

Análisis de Complejidad temporal

|  |  |
| --- | --- |
| *inicio* |  |
| *a=coefficienteN[0]* | 1 |
| *b=coefficenteN[1]* | 1 |
| *c=coeficienteN[2]* | 1 |
| *x1=(-b+ SQR((b\*b) – (4\*a\*c)))/2\*a* | 1 |
| *x2=(-b- SQR((b\*b)-(4\*a\*c)))/2\*a* | 1 |
| *Fin*  *t(n)=* | 5= o(1) |

La complejidad temporal del método es T(n)=5=O(1)

*Análisis de complejidad Espacial*

|  |  |  |
| --- | --- | --- |
| Tipo | Variable | cantidad de valores atómicos |
| Entrada y Salida |  |  |
| Auxiliar | a  b  c  x1  x2 | 1  1  1  1  1 |

Complejidad Espacial totalT(5)= 5=o(1)

**Especificación del problema:**

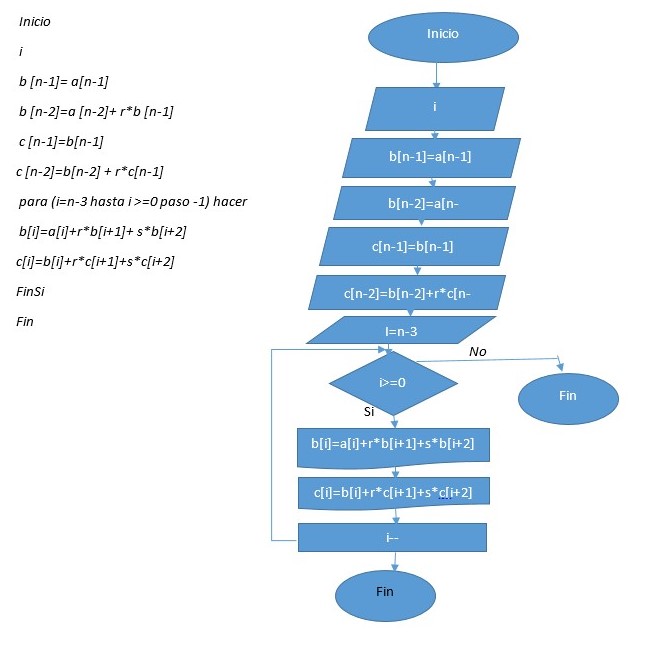
**Problema:** Raíces de un polinomio de grado menor o igual a 10

**Entradas:** conocer los coeficientes de las constantes desde n1..n2...hasta n<=10

**Salida:** Una o varias raíces reales

**Método Bairstow**: Consta de dos métodos:

*División Derivada:*



Complejidad Espacial:

|  |  |
| --- | --- |
| inicio |  |
| i | 1 |
| b [n-1]= a[n-1] | 1 |
| *b [n-2]=a [n-2]+ r\*b [n-1]* | 1 |
| *c [n-1]=b[n-1]* | 1 |
| *c [n-2]=b[n-2] + r\*c[n-1]* | 1 |
| *para (i=n-3 hasta i >=0 paso -1) hacer* | n-1 |
| *b[i]=a[i]+r\*b[i+1]+ s\*b[i+2]* | n-2 |
| *c[i]=b[i]+r\*c[i+1]+s\*c[i+2]* | n-2 |
| *FinSi* |  |
| *Fin* |  |

El algoritmo tiene una complejidad espacial de T(n)=3n=O(n)=n.

Complejidad Temporal

|  |  |  |
| --- | --- | --- |
| Tipo | Variable | Cantidad de valores atómicos |
| Entrada y salida | a  b  c  r  s  n | n  m  x  1  1  1 |
| Auxiliar | i | 1 |

El algoritmo tiene una complejidad espacial de T(n,m.x)=n+m+x+4 = O(n,m,x)

Metodo numero dos:

*Raíces:*

Pseudocódigo del método Diagrama de flujo del método

|  |  |
| --- | --- |
|  |  |

Complejidad temporal

|  |  |
| --- | --- |
| inicio |  |
| *d=r\*r+4\*s* | 1 |
| *Si (d es mayor a 0) entonces* | 1 |
| *re[n-1]= (r+SQR(d))/2.0* | 1 |
| *re[n-2]= (r- SQR(d))/2.0* | 1 |
| *im[n-1]=0* | 1 |
| *im[n-2]=0* | 1 |
| *SiNo* |  |
| *re[n-1]=r/2.0* | 1 |
| *re[n-2]=re[n-1]* | 1 |
| *im[n-1]=SQRT(-d)/2.0* | 1 |
| *im[n-2]=-im[n-1]* | 1 |
| *FinSi* |  |
| *Fin* |  |

La complejidad temporal del método es T(n)=10=0(1).

Complejidad Espacial

|  |  |  |
| --- | --- | --- |
| Tipo | Variable | Cantidad de valores atómicos |
| Entradas y salidas | r  s  re  im  n | 1  1  n  m  1 |
| Auxiliar | d | 1 |

T(n,m)=4+n+m=O(n,m)

La complejidad espacial de algoritmo raíces es O(n,m).