METODO DE LA INGENIERÍA

Fase 1: Identificación del problema

Oracle Corporation es una compañía especializada en el desarrollo de soluciones de nube y locales. En septiembre de 2018 la empresa lanzó al mercado su nuevo desarrollo el Java 11. La primera versión con un JDK denominado LTS.

Hoy en día se encuentran desarrollando una nueva versión que será lanzado en marzo de 2019 en el cual están trabajando en muchos features en cuanto a los servicios que emplea Java.

Una nueva funcionalidad que vendrá es encontrar las raíces de un polinomio lo cual para los desarrolladores es una funcionalidad muy importante y para los ingenieros puede llegar a ser de gran ayuda.

- ### Definición del problema

Se ha requerido el desarrollo de una funcionalidad que encuentre las raíces de un polinomio.

Requerimientos funcionales

La solución del problema:

R1: Requiere generar aleatoriamente polinomios de grado 10, los cuales puedan servir de entrada a los algoritmos que se proponen. Esta opción se desarrolla con el fin de exista la opción de generar polinomios y no necesariamente ingresarlos.

R2: Debe tener una interfaz gráfica que permita ingresar el polinomio que se desea resolver y realizar todo lo propuesto con él. Dicha interfaz debe interactuar con el usuario pidiendo el grado del polinomio y el coeficiente de cada uno de los componente acorde al grado introducido. Por ejemplo, se pide el grado del polinomio y el usuario ingreso 2, entonces se pedirán los coeficiente del grado 2, 1 y el grado 0.

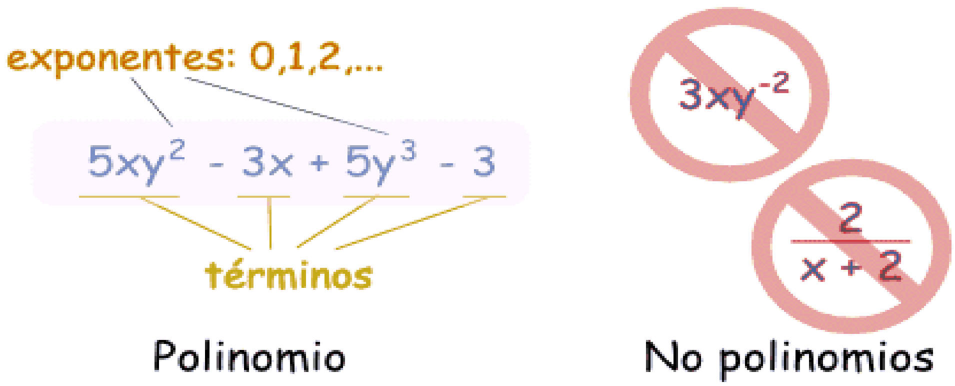
R3: Debe permitir ver al usuario las raíces del polinomio que fue entregado por el previamente sean reales o complejas, o en ciertos casos ambas.

R4: Debe Realizar la solución de las raíces del polinomio.

R5: Si las soluciones del polinomio no son números reales se debe lanzar un mensaje con el mismo y se deben dar las solución con números complejos.

Fase 2: Recopilación de la información

POLINOMIOS



En matemáticas, un polinomio (del latín polynomium, y este del griego, πολυς polys ‘muchos’ y νόμος nómos ‘regla’, ‘prescripción’, ‘distribución’) es una expresión algebraica constituida por una suma finita de productos entre variables (valores no determinados o desconocidos) y constantes (números fijos llamados coeficientes), o bien una sola variable. Las variables pueden tener exponentes de valores definidos naturales incluido el cero y cuyo valor máximo se conocerá como grado del polinomio. En términos más simples, un polinomio se toma como una suma de monomios, pero un monomio también se toma como un polinomio.

Es frecuente el término polinómico (ocasionalmente también el anglicismo polinomial), como adjetivo, para designar cantidades que se pueden expresar como polinomios de algún parámetro, como, por ejemplo: tiempo polinómico, etc.

Los polinomios son objetos muy utilizados en matemáticas y en ciencia. En la práctica, son utilizados en cálculo y análisis matemático para aproximar cualquier función derivable; las ecuaciones polinómicas y las funciones polinómicas tienen aplicaciones en una gran variedad de problemas, desde la matemática elemental y el álgebra hasta áreas como la física, química, economía y las ciencias sociales.

Grado de un polinomio

Se define el grado de un monomio como el exponente de su variable. El grado de un polinomio es el del monomio de mayor grado, y se denota por gr(p).

Ejemplos

$P(x) = 2$, polinomio de grado cero (el polinomio solo consta del término independiente).

$P(x) = 3x + 2$, polinomio de grado uno.

$P(x) = 3x^2 + 2x$, polinomio de grado dos.

P(x) = 2x3+ 3x + 2, polinomio de grado tres.

P(x) = 4x4+ 4x + 2, polinomio de grado cuatro.

P(x) = 2x5+ 3x + 1, polinomio de grado cinco.

Convencionalmente se define el grado del polinomio nulo como $-\infty$

En particular los números son polinomios de grado cero.

Polinomio cero

Es el 0, tiene grado -1. Actúa de elemento neutro aditivo: $p(x) +0= p(x)$, para cualquier $p(x)$.

Polinomio de grado cero

Es aquel que no lleva la indeterminada. Son los elementos no nulos de conjuntos numéricos correspondientes.

Raíces de un polinomio

Las raíces de un polinomio son números tales que hacen que un polinomio valga cero. Podemos decir también que las raíces enteras de un polinomio de coeficientes enteros serán divisores del término independiente. Cuando resolvemos un polinomio igualándolo a cero obtenemos como soluciones las raíces del polinomio. Como propiedades de las raíces y factores de los polinomios podemos decir que los ceros o raíces de un polinomio son por los divisores del término independiente pertenecientes al polinomio. Entonces a cada raíz por ejemplo del tipo x = a le correspondería un binomio del tipo (x-a). Se puede expresar un polinomio en factores si lo escribimos como producto de todos los binomios que tengamos del tipo (x-a) que sean correspondientes a las raíces, x=a, que obtengamos.

Ejemplo:

$$x^2 - 5x + 6 = (x - 2) \cdot (x - 3)$$

Debemos tener en cuenta que la suma de los exponentes de los binomios es igual al grado del polinomio, también tengamos en cuenta que todo polinomio que no tenga término independiente admitirá como raíz x=0, en otra forma, admitirá como factor x. Por ejemplo:

$$x^2 + x = x \cdot (x + 1)$$

Raíces: $x = 0$ y $x = -1$

Llamaremos “primo” o “Irreducible” a un polinomio cuando no hay posibilidad de descomponerlo en factores. Observemos el siguiente ejemplo:

$$P(x) = x^2 + x + 1$$

ara adentrarnos más en el tema es necesario tener claro el teorema fundamental del álgebra, el cual fundamenta que un polinomio en una variable no constante y de coeficientes complejos, tiene tantas raíces como su grado, ya que las raíces se cuentan con sus multiplicidades. Se está afirmando con esto que cualquier ecuación algebraica de grado n posee puntualmente n soluciones complejas. Un polinomio de grado n tiene como máximo n raíces reales. Las raíces complejas de un polinomio de coeficientes reales continuamente se presentan de a pares, un polinomio de grado impar tiene mínimamente una raíz real. Hay que tener en cuenta también que un polinomio puede no poseer raíces reales. Un polinomio que tenga raíces reales y distintas es uno de los casos más simples que podemos encontrar. Por ejemplo, en el siguiente polinomio en que se puede corroborar que sus raíces son, 3; 2 y -1.

$$x^3 - 4x^2 + x + 6$$

En caso de que los coeficientes del polinomio sean complejos las raíces complejas no estarán necesariamente relacionadas. Los polinomios pueden tener entonces x^2+1

raíces complejas, y sus respectivas conjugadas. Por ejemplo un polinomio :
Tiene una raíz compleja y su correspondiente conjugada. Para calcular una raíz compleja determinaremos su parte real, ya que la parte imaginaria, menor que cero se obtiene a partir de su módulo y de su parte real.

Raíces de un polinomio: Método de Graeffe

Sea el polinomio

$a_0x^n+a_1x^{n-1}+a_2x^{n-2}+a_3x^{n-3}+ \ldots a_{n-1}x+a_n=0$ (1)

Hacemos el polinomio más simple dividiendo todos los coeficientes por el primer término de modo que a_0 es siempre 1. Supongamos que sus raíces reales y distintas son

$-r_1, -r_2, -r_3, \ldots -r_n$

Al elevar al cuadrado el polinomio y agrupar los términos se obtiene un polinomio de grado 2n

$a_0^2(x^n)^2-(a_1^2-2a_2a_0)(x^{n-1})^2+(a_2^2-2a_1a_3+2a_4a_0)(x^{n-2})^2-(a_3^2-2a_2a_4+2a_1a_5-2a_6a_0)(x^{n-3})^2+ \ldots=0\}$ (2)

Cuyas raíces serán

$-r_1^2,-r_2^2,-r_3^2 \ldots -r_n^2$

Hemos construido así una nueva ecuación cuyas raíces son numéricamente iguales a los cuadrados de las raíces de la ecuación original. Repitiendo el proceso, se pueden obtener ecuaciones cuyas raíces sean numéricamente iguales a las potencias cuarta, octava, decimosexta, etc. de las raíces de la ecuación original. El efecto de este proceso de elevar al cuadrado es el de producir ecuaciones cuyas raíces están cada vez más separadas. Por ejemplo, si dos raíces de la ecuación original están entre sí como 5 : 4 sus potencias 128 están en la razón $5^{128} : 4^{128}$, o sea, $2.54\ 10^{12}$: 1, lo que es muy deseable ya que las ecuaciones cuyas raíces están muy separadas se pueden resolver rápidamente con exactitud considerable. Supóngase ahora, que reiterando el proceso de elevación al cuadrado se llega a un polinomio

$\alpha_0(x^n)^{2^m}+\alpha_1(x^{n-1})^{2^m}+\alpha_2(x^{n-2})^{2^m}+\alpha_3(x^{n-3})^{2^m}+ \ldots=0$ (3)

donde m es el número de veces que se repite el proceso de elevación al cuadrado. Así, si se repite siete veces el proceso de elevación al cuadrado, $2^m = 2^7 = 128$ sería el exponente al que estarían elevados las sucesivas potencias x^n,

x^{n-1}, x^{n-2}, ... del polinomio. Sus raíces serán las del polinomio original elevadas al exponente 2m.

$-r_1^{2m}, -r_2^{2m}, -r_3^{2m}, \ldots -r_n^{2m}$

Por las relaciones conocidas entre raíces y coeficientes del polinomio, se tiene que

$\alpha_0=1$

$\alpha_1=(\text{suma de las raíces})=r_1^{2m}+r_2^{2m}+\ldots+r_n^{2m}$

$\alpha_2=(\text{suma de las raíces tomando dos cada vez})=$

$r_1^{2m}r_2^{2m}+r_1^{2m}r_3^{2m}+\ldots+r_2^{2m}r_3^{2m}+\ldots+r_{n-1}^{2m}r_n^{2m}$

$\alpha_3=-(\text{suma de las raíces tomando tres cada vez})$

$=r_1^{2m}r_2^{2m}r_3^{2m}+r_1^{2m}r_2^{2m}r_4^{2m}+\ldots+r_2^{2m}r_3^{2m}r_4^{2m}+\ldots+r_{n-2}^{2m}r_{n-1}^{2m}r_n^{2m}$

$\alpha_n=(-1)^n(\text{producto de todas las raíces})=r_1^{2m}r_2^{2m}r_3^{2m}\ldots r_n^{2m}$

En la suposición de que

$|r_1|>|r_2|>|r_3|> \ldots |r_n|$

y de que 2m es grande por ejemplo 128 ó 256, se cumplirá que

$|r_1|^{2m}>>>|r_2|^{2m}>>>|r_3|^{2m}>>> \ldots |r_n|^{2m}$

donde el símbolo >>> indica mucho mayor que. Las relaciones entre coeficientes y raíces quedarán simplificadas con gran aproximación a las expresiones.

$\alpha_0=1$

$\alpha_1=r_1^{2m}$

$\alpha_2=r_1^{2m}r_2^{2m}$

$\alpha_3=r_1^{2m}r_2^{2m}r_3^{2m}$

$\alpha_n=r_1^{2m}r_2^{2m}r_3^{2m}\ldots r_n^{2m}$

Así, el módulo de r_1 se puede hallar extrayendo la raíz 2m-ésima de α_1 . De la segunda ecuación se obtiene r_2, y así sucesivamente. La fórmula para obtener el módulo de la raíz r_i es

$|r_i|=\sqrt[2m]{\frac{\alpha_i}{\alpha_{i-1}}}$

En la práctica, hallamos el logaritmo de r_i, y luego, calculamos el antilogaritmo del resultado obtenido, de este modo se obtiene el valor absoluto de la raíz r_i.

$\log|r_i| = \log \alpha_i - \log \alpha_{i-1} 2m$ (4)

Para determinar el signo, se halla el valor del polinomio original para los valores ri, y -ri, uno de los dos hará que dicho valor sea próximo a cero y por tanto, será la raíz buscada.

Fase 3: Búsqueda de soluciones creativas

- La generación de las ideas se desarrollaron de forma conjunta planeando cuales serían las posibles soluciones al problema, donde aceptamos todo tipo de idea, pero con el fin de que su uso fuese lógico en la solución, sin importar si era el más optimo, el más eficiente o algún otro criterio.

1.Buscar algoritmos existentes que nos permitan facilitar la implementación con los conocimientos ya obtenidos en APO2 como lo son listas y el uso de árboles binarios, guardando la información de los polinomios en estas estructuras.

2.Implementar nuestros propios algoritmos, teniendo en cuenta las reglas para dar solución a un polinomio.

3.Implementar el algoritmos de Graeffe para hallar los ceros de un polinomio, tanto imaginarios como reales.

4.Mediante el uso de algoritmos de ordenamiento organizar cada uno de los polinomios acorde a su grado.

5.Utilizar el concepto de números complejos para poder dar solución a un polinomio cuyas raíces no sean números reales, utilizando su definición y llevándola a código a través de métodos simples.

6.Realizar división de polinomios con la Regla de Ruffini de forma que al tener un polinomio de grado n se dice que tiene n raíces enteras, por lo cual se factoriza como en el siguiente ejemplo de polinomio de grado 4.

$ax^4+bx^3+cx^2+dx+e = a(x-x_1)(x-x_2)(x-x_3)(x-x_4)$

7.Desarrollar un solo algoritmo que me resuelva los casos donde pueda sacar factor común para luego hallar las raíces del polinomio. Este puede ser a través del uso de algoritmos repetitivos(while-for).

Fase 4: Transición de las Ideas a los Diseños Preliminares

Las siguientes ideas las descartaremos tomando como base de criterio el desarrollo por completo y de forma efectiva lo que se nos esta pidiendo en este caso con el desarrollo de la factorización de polinomios para hallar sus raíces, no obstante en cada una de las ideas descartadas que daremos a continuación argumentaremos por qué se tomó la decisión.

<u>Alternativa 1: Uso de algoritmos ya conocidos y utilizados anteriormente</u>

-El uso de Listas es totalmente absurdo y aunque puede llegar a ser interesante no se empleará su uso como fundamento del desarrollo de la solución.

-Por el mismo argumento de descarta la utilización de arboles binarias, ya que este método o vía podrá servir en algo, pero en nada al fin que es encontrar las raíces de un polinomio de grado n.

<u>Alternativa 2: Creación de propios algoritmos</u>

-Este método puede llegar a ser muy bueno, pero debido a que ya hay métodos que desarrollan una solución a lo solicitado entonces es para nada eficiente emplear esta idea.

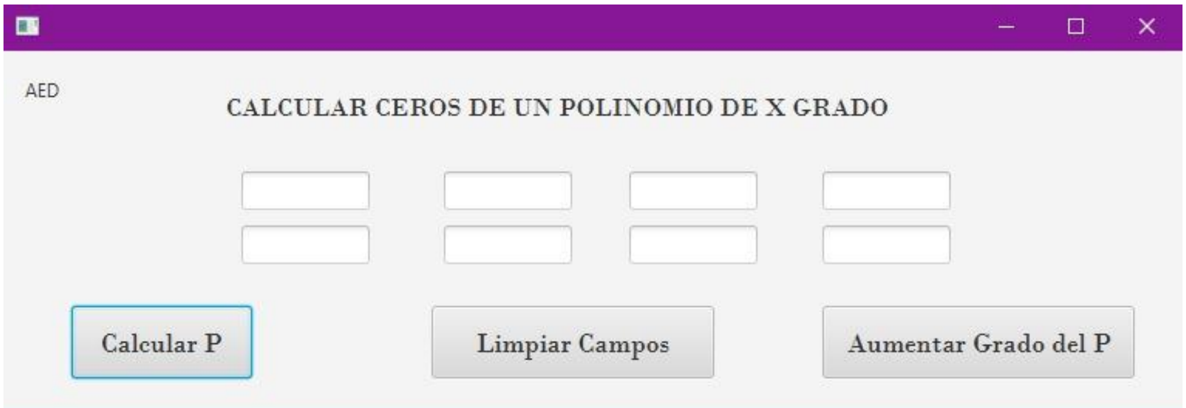
-El uso de recursos como el tiempo es exageradamente usado si se desarrolla esta idea, es por esa razón que es descartada.

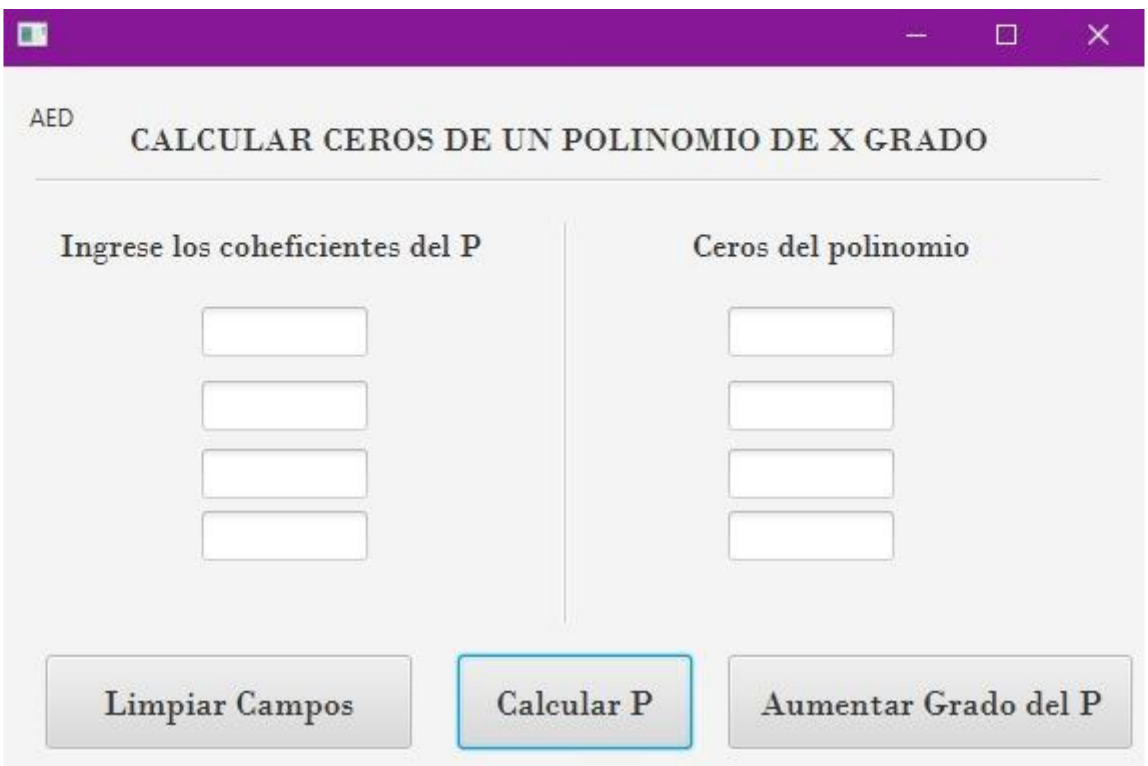
<u>Alternativa 4: Uso de métodos de ordenamiento</u>

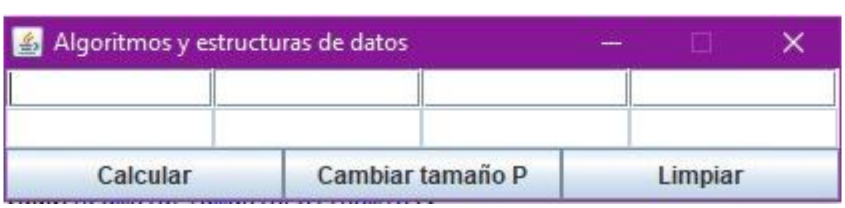
-El uso de estos algoritmos ante este caso es obsoleto por más que podamos llegar a organizar los polinomios según sus grados, pero como en casos anteriores esto no me emplea el fundamento de mi problema que es hallar las raíces de los polinomios.

Diseños Preliminares

-Bosquejos de la interfaz del programa:







Pseudocódigos

valorPolinomio(Complejo x) throws IOException

y = new Complejo;

Para i=0 hasta i<n+1

y= Complejo.suma(y, Complejo.producto(a[o][i], Complejo.potencia(x,(n-1))));

aumentando i++;

retorna y;

unaRaízCompleja()

suma ← 0.0;

Para i=0 hasta i<numReales

suma += raicesReales[i];

aumentando i++;

double u, v;

u ← a[0][1]+suma / 2;

v ← Math.sqrt(moduloComplejas[0]*moduloComplejas[0]-u*u);

raicesComplejas[0] ← new Complejo(u,v);

raicesComplejas[1] ← new Complejo(u, -v);

hallarRaices()

Para i = 0 hasta i < n+1 i++

Si cambiaSigno (i)

i ++

logaritmo = Math.Log(a[m][i])-2*Math.Log(a [m-1][i])

Si Math.abs(logaritmo) < CERO

raizRealSimple(i)

Si no

raizRealDoble (i)

i ++

Si numComplejas == 1

UnaRaizCompleja()

SI numCompleja==2

dosRaicesComplejas

Fase 5: Evaluación y Selección de la Mejor Solución

Criterios

Estos son los principios por los cuales serán evaluadas las ideas y donde escogeremos las apropiadas para el desarrollo de la solución del problema de forma definitiva.

Criterio A: Complejidad

- [8] Complejidad constante
- [7] Complejidad logarítmica
- [6] Complejidad raíz
- [5] Complejidad lineal
- [4] Complejidad nlogn
- [3] Complejidad polinómica
- [2] Complejidad exponencial

- [1] Complejidad factorial

Criterio B: Eficiencia

- [3] Muy eficiente
- [2] Eficiente
- [1] Nada eficiente

Criterio C: Facilidad implementación

- [2] Fácil
- [1] Difícil

Evaluación

Evaluando los criterios anteriores en las alternativas que se mantienen, obtenemos la siguiente tabla:

	Criterio A	Criterio B	Criterio C	Total
Alternativa 3: Método de Graeffe	Polinómica – 3	Muy Eficiente- 3	Fácil- 2	8
Alternativa 5: Concepto raíces complejas	Lineal - 5	Eficiente - 2	Difícil - 1	8

Alternativa 7: Uso de estructuras iterativas	Polinómica - 3	Eficiente - 2	Fácil - 2	7

Selección

De acuerdo con la evaluación anterior de se debe seleccionar las Alternativas 3 y 5, ya que obtuvieron la mayor puntuación de acuerdo a los criterios definidos.