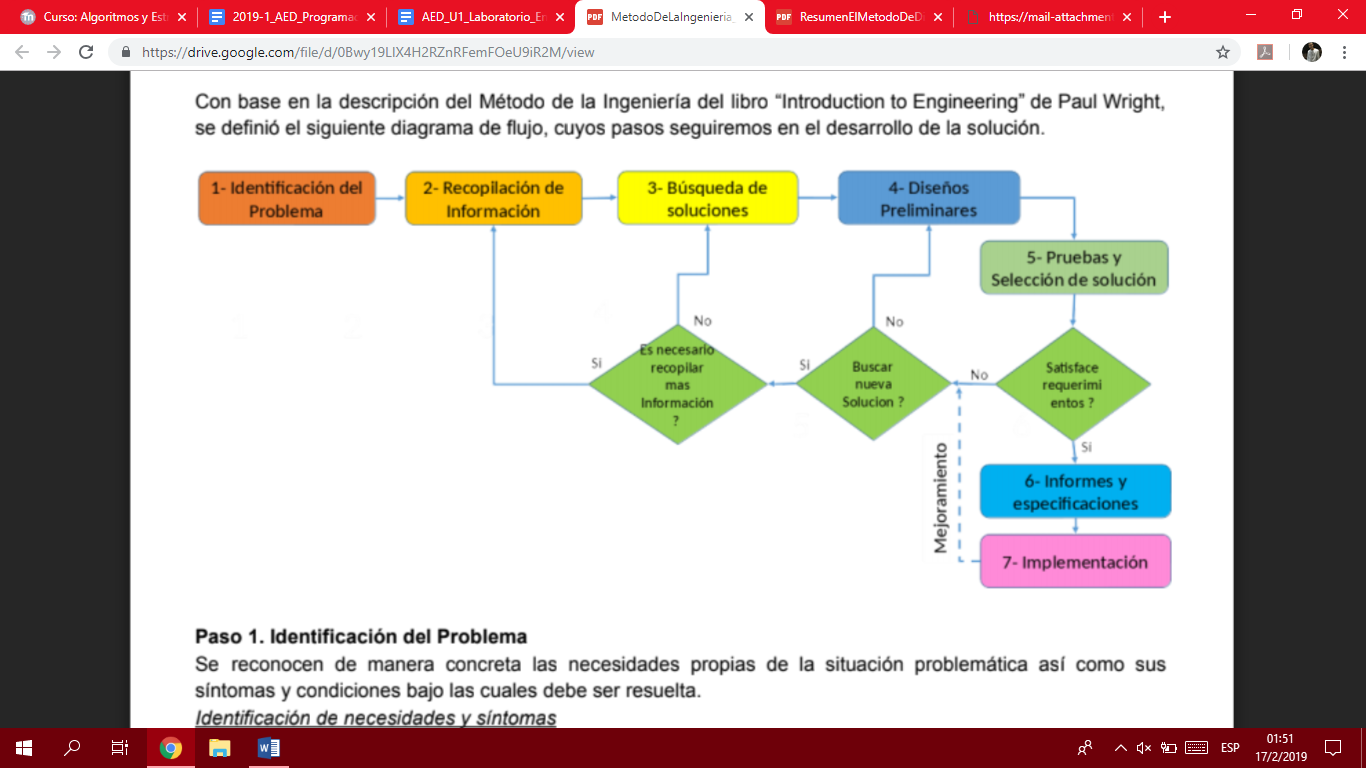
**Contexto problemático:**

La compañía que ocupa el primer lugar en la categoría de bases de datos Oracle Corporation, especializada en el desarrollo de soluciones de nube y locales requiere encontrar las raíces de un polinomio.

**Desarrollo de la solución:**

Para resolver el caso problemático anterior se eligió el Método de la Ingeniería del libro “Introduction to Engineering” de Paul Wright.

Los pasos para conseguir el desarrollo de la solución se describen en el siguiente diagrama de flujo.



**Paso 1. Identificación del problema:**

El problema que nos presenta el cliente es encontrar las raíces de un polinomio, ya que esta funcionalidad puede ser de gran ayuda para estudiantes de ingeniería y de aquí se deriva:

* Ofrecer mínimo dos algoritmos para encontrar las raíces de los polinomios.
* Generar aleatoriamente polinomios hasta de grado 10.
* Poder encontrar las raíces de los polinomios ingresados y generados aleatoriamente.

**Paso 2. Recopilación de información:**

Para poder dar solución al problema, primero debemos entrar en materia con el tema pedido como requerimiento, en este caso, las raíces de un polinomio, recopilamos la siguiente información con el fin de aclarar todas las dudas propuestas por nuestro grupo de trabajo respecto al caso problemático:

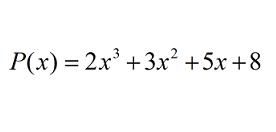
**Expresión algebraica:**

Cadena de símbolos matemáticos que indican una cantidad finita de operaciones básicas entre funciones elementales, como raíces, exponenciales, logaritmos, funciones trigonométricas y también composiciones de dichas funciones.

**Clasificación de las expresiones algebraicas:**

* **Monomio:** Un monomio es una expresión algebraica en la que las únicas operaciones que aparecen entre las variables son el producto y la potencia de exponente natural.
* **Polinomio:** En matemáticas, un polinomio (del latín polynomium)​ 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.

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.



(Ejemplo y representación gráfica de un polinomio)

**Valor numérico de una expresión algebraica:**

Si en una expresión algebraica se sustituyen las variables por números y se efectúan las operaciones indicadas, el valor resultante (si existe) recibe el nombre de valor numérico de la expresión algebraica.

**Grado de un monomio:**

Se denomina grado de un monomio (en el cual sus factores literales aparezcan con exponentes enteros no negativos), a la suma de los exponentes de las variables que contengan.

**Grado de un polinomio:**

Se denomina grado de un polinomio al mayor de los grados de los términos (monomios) que lo componen.

**Raíces de un polinomio:**

Las raíces de un polinomio son números tales que hacen que un polinomio valga cero (Valor numérico de la expresión algebraica sea igual a 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.

**Notas interesantes:**

* Un polinomio tiene tantas raíces como su grado lo indique.
* Las raíces complejas de un polinomio se presentan de a pares.
* Un polinomio puede no poseer raíces reales.

**Fuentes:**

<https://www.ecured.cu/Expresi%C3%B3n_algebraica>

<https://www.ecured.cu/Polinomio>

<https://matematica.laguia2000.com/general/raices-de-un-polinomio>

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

**Paso 3. Búsqueda de soluciones creativas:**

En este paso, visitamos varios sitios web con mucha información donde encontramos varias alternativas que de una u otra forma dan solución al problema, estas son:

**Alternativa 1:**

**Método de la bisección:** Es el método más elemental y antiguo para determinar las raíces de una ecuación. Está basado directamente en el teorema de Bolzano que consiste en partir de un intervalo [] tal que < 0, por lo que sabemos que existe, al menos, una raíz real. A partir de este punto se va reduciendo el intervalo sucesivamente hasta hacerlo tan pequeño como exija la precisión que hayamos decidido emplear.

**Alternativa 2:**

**Método de las aproximaciones sucesivas o punto fijo:** Dada la ecuación , el método de las aproximaciones sucesivas reemplaza esta ecuación por una equivalente, , definida en la forma . Para encontrar la solución, partimos de un valor inicial y calculamos una nueva aproximación . Reemplazamos el nuevo valor obtenido y repetimos el proceso. Esto da lugar a una sucesión de valores {}, que si converge, tendrá como límite la solución del problema.

**Alternativa 3:**

**Método de Newton-Raphson:** Este método parte de una aproximación inicial y obtiene una aproximación mejor dada por la fórmula . La expresión anterior puede derivarse a partir de un desarrollo de serie de Taylor. El método de Newton consiste en una linealización de la función, es decir, se reemplaza por una recta tal que contiene al punto y cuya pendiente coincide con la derivada de la función en el punto, . La nueva aproximación a la raíz, , se obtiene de la intersección de la función linear con el eje X de ordenadas.

**Alternativa 4:**

**Método de la secante:** El método de la secante parte de dos puntos (y no sólo uno como el método de Newton) y estima la tangente (es decir, la pendiente de la recta) por una aproximación de acuerdo con la expresión . En general, el método de la secante presenta las mismas ventajas y limitaciones que el método de Newton-Raphson explicado anteriormente.

**Alternativa 5:**

**Método de Steffensen:** El método de Steffensen presenta una convergencia rápida y no requiere, como en el caso del método de la secante, la evaluación de derivada alguna. Presenta, además, la ventaja adicional de que el proceso de iteración sólo necesita un punto inicial. Este método calcula el siguiente punto de iteración a partir de la expresión .

**Alternativa 6:**

**Método de la falsa posición:** El método de la falsa posición pretende conjugar la seguridad del método de la bisección con la rapidez del método de la secante. Este método, como en el método de la bisección, parte de dos puntos que rodean a la raíz , es decir, dos puntos y tales que . La siguiente aproximación, , se calcula como la intersección con el eje X de la recta que une ambos puntos. La asignación del nuevo intervalo de búsqueda se realiza como en el método de la bisección: entre ambos intervalos, [] y [], se toma aquel que cumpla . La elección guiada del intervalo representa una ventaja respecto al método de la secante ya que inhibe la posibilidad de una divergencia del método. Por otra parte, y respecto al método de la bisección, mejora notablemente la elección del intervalo (ya que no se limita a partir el intervalo por la mitad).

**Alternativa 7:**

**Algoritmo de Horner:** Método numérico debido al británico W.G. Horner (1786-1837) que, a base de aproximaciones sucesivas, permite calcular las soluciones reales de cualquier ecuación algebraica con coeficientes reales, con tanta aproximación como se desee. Estos son los procedimientos de este algoritmo:

* Se escribe los coeficientes del dividendo en una fila con su propio signo
* Se escribe los coeficientes del divisor en una columna a la izquierda del primer término del dividendo; el primero de ellos con su propio signo y los restantes con signo cambiado.
* El primer término del dividendo se divide entre el primer término del divisor, obteniéndose el primer término del cociente.
* Se multiplica este término del cociente solamente por los términos del divisor a los cuales se cambió de signo, colocándose los resultados a partir de la segunda fila, corriendo un lugar hacia la derecha.
* Se reduce la siguiente columna y se coloca el resultado en la parte superior para dividirlo entre el primer coeficiente del divisor y obtener el segundo término del cociente.
* Se multiplica este cociente por los términos del divisor a los cuales se cambió de signo, colocándose el resultado en la tercera fila y corriendo un lugar hacia la derecha.
* Se continuaría este procedimiento hasta obtener el término debajo del último término del dividendo, separando inmediatamente los términos del cociente y resto.
* Para obtener los coeficientes del residuo se reducen directamente cada una de las columnas que pertenecen.

**Alternativa 8:**

**Regla de Ruffini:** En matemáticas, la regla de Ruffini facilita el cálculo rápido de la división de cualquier polinomio entre un binomio de la forma . Descrita por Paolo Ruffini en 1809, es un caso especial de división sintética (una división de polinomios en donde el divisor es un factor lineal). El Algoritmo de Horner para la división de polinomios utiliza la regla de Ruffini (también se la conoce como Método de Horner o Algoritmo de Ruffini-Horner). La regla de Ruffini permite así mismo localizar las raíces de un polinomio y factorizarlo en binomios de la forma (siendo r un número entero) si es coherente.

**Nota:**

La mayoría de los métodos utilizados para el cálculo de las raíces de una ecuación son iterativos y se basan en modelos de aproximaciones sucesivas. Estos métodos trabajan del siguiente modo: a partir de una primera aproximación al valor de la raíz, determinamos una aproximación mejor aplicando una determinada regla de cálculo y así sucesivamente hasta que se determine el valor de la raíz con el grado de aproximación deseado.

**Fuentes:**

<https://www.uv.es/~diaz/mn/node17.html>

<https://juncotic.com/metodo-de-horner-algoritmos-antiguos/>

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

<https://ekuatio.com/la-regla-de-ruffini/>

**Paso 4. Transición de las ideas a los diseños preliminares:**

Revisando teórica y prácticamente cada una de las alternativas estos fueron los detalles de cada una:

**Alternativa 1: Método de la bisección:**

* El método de bisección es muy seguro para garantizar convergencia. Si f es una función continua en el intervalo [a, b] y f(a)f(b) < 0, entonces este método converge a la raíz de f.
* La bisección converge linealmente, por lo cual es un poco lento. Sin embargo, se garantiza la convergencia si f(a) y f(b) tienen distinto signo.
* Si existieran más de una raíz en el intervalo entonces el método sigue siendo convergente pero no resulta tan fácil caracterizar hacia qué raíz converge el método.
* El algoritmo tendrá un fallo en el caso que la función no sea continua dentro del intervalo proporcionado.
* Un caso frontera seria cuando la función converja a 0 en un intervalo que incluya infinito, esto causara que el algoritmo este eternamente aproximando la solución de la raíz, pero nunca llegando a ser específicamente 0. Para estos casos, los parámetros opcionales son muy útiles, pues podemos dar una tolerancia y aceptar una respuesta que no sea 0 pero si muy cercano o poner un máximo número de iteraciones y retornar el valor que el algoritmo pudo calcular hasta ese momento.
* El método de bisección tiene limitaciones sobre otros métodos numéricos para obtener raíces de ecuaciones no lineales, sin embargo, da resultados aproximados para la simplicidad del algoritmo.

**Alternativa 2: Método de las aproximaciones sucesivas o punto fijo:**

* En la práctica, en lugar de calcular a priori el número de iteraciones a realizar, se va estimando en cada iteración la distancia del valor en ella hallado a la solución exacta. Esta estimación se realiza simplemente evaluando la diferencia entre las dos últimas aproximaciones halladas que, cuando g(x) es una contracción, son un indicador de la cercanía a la solución exacta.
* En los algoritmos que recojan el método, este control de la convergencia debe acompañarse con la limitación del número de iteraciones a realizar en previsión de los casos en los que, no siendo g(x) una contracción, el método no converja.
* En ocasiones la aplicación g(x) con la que se intente aplicar el método de aproximaciones sucesivas a la ecuación x = g(x) conducirá a un proceso que converge muy lentamente por tener su constante de Lipschit próxima a 1. En esas ocasiones será conveniente modificar la ecuación equivalente convirtiéndola en otra de la forma x = h(x) en la que h(x) tenga mejores propiedades de cara a la convergencia del método hacia la solución x’

**Alternativa 3: Método de Newton-Raphson:**

* El método de Newton es muy rápido y eficiente ya que la convergencia es de tipo cuadrático (el número de cifras significativas se duplica en cada iteración).
* Sin embargo, la convergencia depende en gran medida de la forma que adopta la función en las proximidades del punto de iteración.
* Este método algunas veces obviamente no es capaz de alcanzar la convergencia o bien converge hacia un punto que no es un cero de la ecuación.
* El error es proporcional al cuadrado del error anterior, es decir, si el error , el siguiente error  es proporcional a  y así sucesivamente.
* Con lo que podríamos decir que en cada iteración aproximadamente se duplica el número de dígitos correctos.
* Sin embargo, algunas veces el método de Newton no converge, sino que se encicla. Esto puede ocurrir, por ejemplo, si no hay raíz real, si la raíz es un punto de inflexión o si la aproximación inicial está muy lejos de la raíz buscada y el proceso de aproximación cae en un ciclo.

**Alternativa 4: Método de la secante:**

* El principal inconveniente del método de Newton estriba en que requiere conocer el valor de la primera derivada de la función en el punto. Sin embargo, la forma funcional de f(x) dificulta en ocasiones el cálculo de la derivada. En estos casos es más útil emplear el método de la secante.
* En general, el método de la secante presenta las mismas ventajas y limitaciones que el método de Newton-Raphson explicado anteriormente.
* En caso de que la aproximación inicial sea demasiado lejana o la raíz no sea simple, este método no asegura la convergencia y tiene un comportamiento similar al de Newton-Raphson.
* Si comparamos el método de Newton-Raphson con el método de la secante, vemos que el método de Newton-Raphson converge más rápido. Sin embargo, el método de Newton-Raphson requiere la evaluación de ambos f y su derivada en cada paso, mientras que el método de la secante sólo requiere la evaluación de f. Por lo tanto, el método de la secante puede muy bien ser más rápido en la práctica.

**Alternativa 5: Método de Steffensen:**

* El método de Steffensen presenta una convergencia rápida y no requiere, como en el caso del método de Newton, la evaluación de derivada alguna.
* Presenta, además, la ventaja adicional de que el proceso de iteración sólo necesita un punto inicial.
* Otra ventaja del método de Steffensen es que (al igual que el de Newton) tiene convergencia cuadrática. Es decir, ambos métodos permiten encontrar las raíces de una función f "rápidamente" (en este caso rápidamente significa que, en cada iteración, el número de dígitos correctos en la respuesta se duplica). Pero la fórmula para el método de Newton requiere la evaluación de la derivada de la función, el método de Steffensen no, por lo que este último puede ser programado para una función genérica, mientras que la función cumpla la restricción mencionada anteriormente.
* El precio de la convergencia rápida es una doble evaluación de la función: tanto como deben ser calculadas, lo que podría llevar un tiempo considerable dependiendo de la función f. Por comparación, el método de la secante sólo necesita una evaluación de la función por cada paso, así que con dos evaluaciones de la función del método de la secante se pueden hacer dos pasos, y esos dos pasos aumentan el número de dígitos correctos en un factor de 1,6. En un solo paso de tiempo el método de Steffensen (o de Newton) aumenta los dígitos correctos en un factor de 2, lo que es sólo un poco mejor.
* Al igual que el método de Newton y otros métodos cuadráticamente convergentes, la debilidad fundamental en el método de Steffensen es la elección del valor inicial . Si el valor de no está "lo suficientemente cerca" de la solución, el método puede fallar y la secuencia de valores o bien puede oscilar entre dos valores, o bien divergir hacia infinito (ambas alternativas pueden suceder).

**Alternativa 6: Método de la falsa posición:**

* La elección guiada del intervalo representa una ventaja respecto al método de la secante ya que inhibe la posibilidad de una divergencia del método.
* Por otra parte, y respecto al método de la bisección, mejora notablemente la elección del intervalo (ya que no se limita a partir el intervalo por la mitad).
* Sin embargo, el método de la falsa posición tiene una convergencia muy lenta hacia la solución. Efectivamente, una vez iniciado el proceso iterativo, uno de los extremos del intervalo tiende a no modificarse.
* Para obviar este problema, se ha propuesto una modificación del método, denominada método de Hamming. Según este método, la aproximación a una raíz se encuentra a partir de la determinación del punto de intersección con el eje X de la recta que une los puntos y si la función es convexa en el intervalo o bien a partir de la recta que une los puntos y si la función es cóncava en el intervalo.
* Se puede demostrar que bajo ciertas condiciones el método de la falsa posición tiene orden de convergencia lineal, por lo que suele converger más lentamente a la solución de la ecuación que el método de la secante, aunque el método de la falsa posición siempre converge a una solución de la ecuación.
* El algoritmo tiene el inconveniente de que, si la función es convexa o cóncava cerca de la solución, el extremo del intervalo más alejado de la solución queda fijo variando únicamente el más cercano, convergiendo muy lentamente.

**Alternativa 7: Algoritmo de Horner:**

* La evaluación usando la forma monomial del polinomio de grado-n requiere al menos n sumas y multiplicaciones, si las potencias se calculan mediante la repetición de multiplicaciones.
* El algoritmo de Horner sólo requiere n sumas y n multiplicaciones. (Minimizar el número de multiplicaciones es lo más deseable porque necesitan mucha carga computacional y son inestables comparadas con la suma).
* Se han demostrado que el algoritmo de Horner es óptimo, de modo que cualquier algoritmo que se use para evaluar un polinomio requerirá como mínimo el mismo número de operaciones. El hecho de que el número de operaciones requeridas es mínimo fue demostrado por Alexander Ostrowski en 1954, y que el número de multiplicaciones es mínimo por Víctor Pan en 1966. Cuando x es una matriz, el algoritmo de Horner no es óptimo.

**Alternativa 8: Regla de Ruffini:**

* El método de Ruffini-Horner es difícilmente explotable si el polinomio posee dos raíces muy cercanas. Ruffini no evoca esta problemática, pero Horner propone un procedimiento especial para estos casos. ​ El método de Horner fue utilizado por los matemáticos De Morgan y J.R. Young.
* Cada vez que hacemos una tabla a partir de los coeficientes del polinomio, obtenemos una raíz y los coeficientes de un polinomio de un grado menor (un polinomio que divide al propio polinomio). De este modo, podemos ir reduciendo el grado del polinomio hasta llegar a uno de segundo grado cuyas raíces sabemos calcular rápidamente.
* En realidad, el método consiste escoger una posible raíz y desarrollar una tabla. Si el último resultado de la tabla es 0, el procedimiento habrá finalizado correctamente. Si no es así, tendremos que probar con otra posible raíz.

**Paso 5. Evaluación y selección de la mejor solución:**

**Criterios:**

* Criterio A. Precisión de la solución. La alternativa entrega una solución:

­ [2] Exacta (se prefiere una solución exacta)

­ [1] Aproximada

* Criterio B. Eficiencia. Se prefiere una solución con mejor eficiencia que las otras consideradas. La eficiencia puede ser:

­ [4] Constante

­ [3] Mayor a constante

­ [2] Logarítmica

­ [1] Lineal

* Criterio C. Completitud. Se prefiere una solución que encuentre todas las soluciones. Cuántas soluciones entrega:

­ [3] Todas

­ [2] Mas de una si las hay, aunque no todas

­ [1] Solo una o ninguna

* Criterio D. Facilidad en implementación algorítmica:

­ [2] Compatible con las operaciones aritméticas básicas de un equipo de cómputo moderno

­ [1] No compatible completamente con las operaciones aritméticas básicas de un equipo de cómputo moderno

**Evaluación:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Criterio A** | **Criterio B** | **Criterio C** | **Criterio D** | **Total** |
| **Newton Raphson** | 1 | 3 | 2 | 1 | 7 |
| **Bisección** | 1 | 3 | 1 | 2 | 7 |

Verificando las alternativas y teniendo en cuenta sus ventajas y desventajas, los métodos seleccionados obtuvieron igual puntuación. Se concluir que, el método de bisección suele ser menos eficiente que el método de newton Raphson, pero es más seguro al momento de garantizar la convergencia.

**Paso 6. Preparación de informes y especificaciones**

Especificación del problema: Método de newton

Problema: Encontrar las raíces de un polinomio

Entrada: Una cadena de caracteres en representación del polinomio. Un p(0) que es un aproximación a la raíz.

Salida: Un valor real.

Especificación del problema: Método de bisección

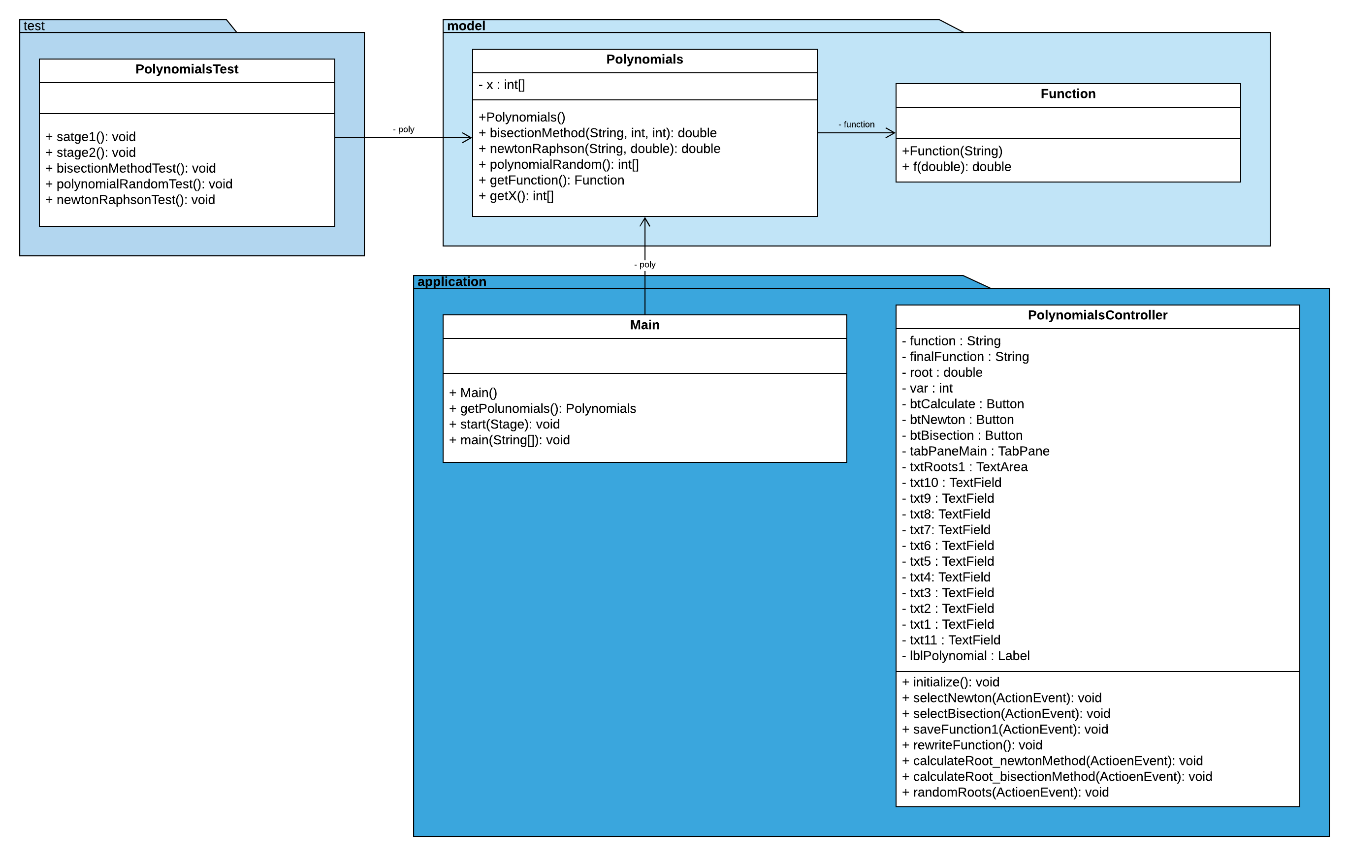
Problema: Encontrar las raíces de un polinomio

Entrada: Una cadena de caracteres en representación del polinomio. Un valor mínimo y otro máximo, que serán evaluados en la función.

Salida: Un valor real.

**Paso 7. Implementación del diseño**

**Diagrama de clases**

****