

Informe Laboratorio: Análisis Numérico

Práctica No. 0

**Estudiante:** Hendrik López

**Código:** 2170129

**Grupo:** B2

*Escuela de ingeniería de sistemas e informática*

*Universidad Industrial de Santander*

20 de abril de 2021

## Introducción

El presente laboratorio ha sido realizado con el propósito de entender y aplicar la aproximación aritmética y el formato de coma flotante, por medio de herramientas de cómputo, como Matlab; para ello, se ha organizado el desarrollo de la actividad en tres fases principales: una serie de preguntas conceptuales y teóricas, que cumplen el papel de introducir al estudiante en el tema en cuestión; problemas manuales que ponen a prueba la capacidad de análisis, aplicando lo explicado previamente y ejercicios aplicados haciendo uso de Matlab.

## Desarrollo

Con el fin de realizar la actividad, es menester dar respuesta a las siguientes preguntas:

* 1. ¿Qué es el error absoluto y el error relativo?

El error absoluto es la diferencia numérica que se tienen en las medidas obtenidas respecto al valor tomado como valor exacto. En síntesis, este puede ser calculado por medio de la siguiente ecuación:

donde es el error absoluto y la diferencia entre las medidas teóricas y las medidas experimentales; de esta manera, se deduce que .

Por ende, el error relativo se interpreta como el cociente entre el error absoluto y el valor teórico p, usando la siguiente ecuación:

donde es el error relativo. Debido a esto, se entiende que

* 1. ¿Cómo calcular los dígitos significativos de un número?

De manera conceptual, los dígitos de un número poseen un grado de importancia al momento de realizar cálculos; debido a esto, existen una serie de consideraciones a la hora de definir dicha relevancia:

* Todos los dígitos diferentes de cero son importantes
* Todos los ceros entre dígitos importantes, son importantes.
  1. El número significativo de cifras está está determinado por el número diferente de cero más cerca de la izquierda.
  2. El número ubicado más hacia la derecha, es el menos significativo
  3. Si no hay punto decimal, el número diferente de cero más hacia la derecha, es el menos significativo.

Lo cual, puede interpretarse con la siguiente ecuación:

donde *d* es considerado el entero positivo más grande dentro de *d* cifras significativas.

1. ¿Qué propiedades posee el orden de aproximación?

El orden de aproximación se refiere a expresiones formales o informales de cuán precisa es una aproximación. Sus propiedades son:

* (Sumar dos órdenes de aproximación de la misma complejidad polinómica, da como resultado él mismo).
* (Sumar dos órdenes de aproximación de distinta complejidad polinómica, da como resultado el orden de aproximación menor).
* (Multiplicar dos órdenes de aproximación da como resultado un nuevo orden de aproximación, donde su exponente es la suma de los exponentes anteriores).

Acto seguido, se resuelven los siguientes ejercicios, aplicando los conceptos explicados por medio del cuestionario anterior:

* 1. Use aproximación aritmética de tres dígitos para calcular las siguientes sumatorias:

El procedimiento es:



El procedimiento es:

1. Mejorar la ecuación cuadrática; se debe asumir que y , teniendo en cuenta la ecuación . Las raíces pueden ser calculadas usando:

Demuestre que las raíces pueden ser calculadas por medio de las siguientes ecuaciones:

En este caso, la manera de llegar a estas expresiones, es por medio de la racionalización. Para el caso de x1, el proceso es:

El proceso es el mismo en el caso de x2:

## Anexos

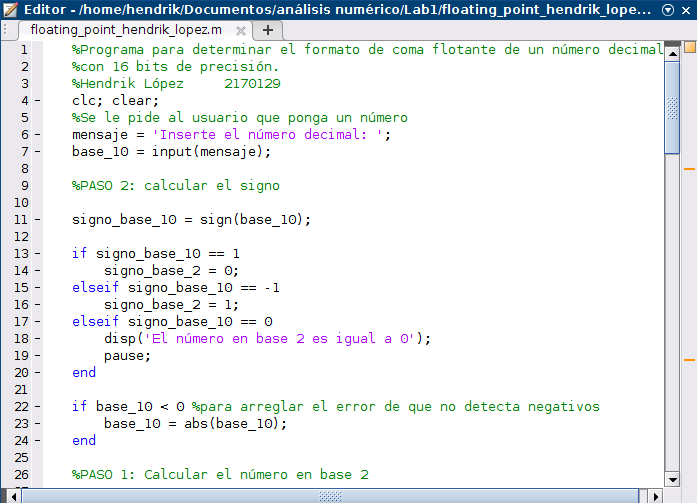
En la presente sección, se presentan los ejercicios aplicados por medio de Matlab, además de capturas del código explicado por medio de comentarios:

1. Crear una función en Matlab llamada *floating\_point\_function\_hendrik\_lopez()* que determine el formato de coma flotante para números almacenados en un computador, con 16 bits de precisión, como se muestra en la siguiente figura:

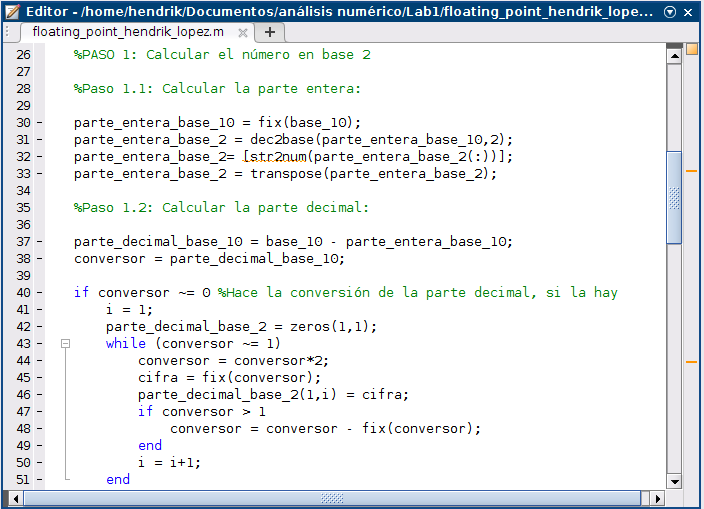


**Figura 1.** Representación del formato de coma flotante

El código que resuelve este problema es el siguiente; primero, se le solicita al usuario que ingrese un número en base 10 con números decimales. Acto seguido, calcula el signo del formato de coma flotante, o define si el número ingresado es 0. Una vez hallado el signo, se calcula el valor absoluto de dicho número, con el fin de evitar errores al manejar números negativos, tal como se muestra en la figura 2:

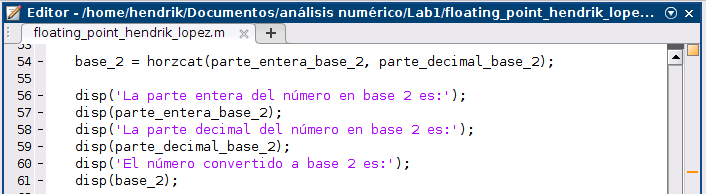
**Figura 2.**  Entrada del número base 10 y cálculo del signo.

Una vez teniendo esta información, el programa procede a hacer la conversión del número en base 10 a base 2, como se puede observar en la figura 3:

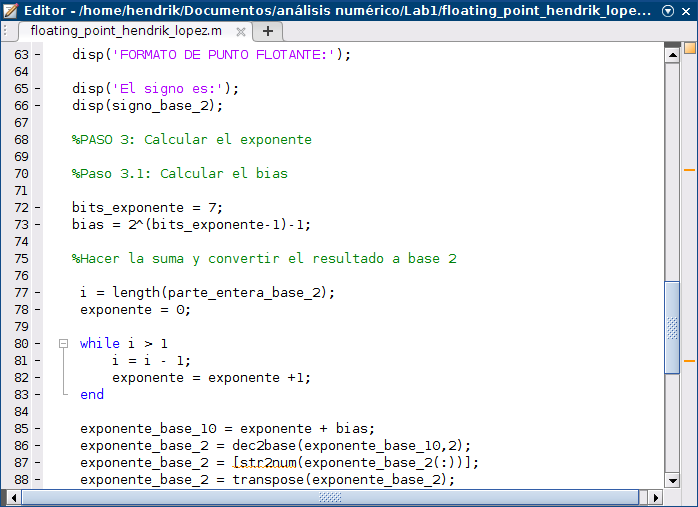


**Figura 3.**  Conversión del número en base 10 a base 2.

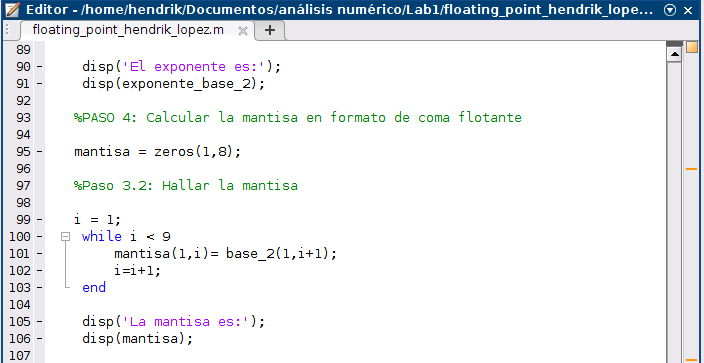
Una vez se tienen las partes enteras y decimales del número en base 2, estas se concatenan en un único vector, con el propósito de visualizar el número completo; este proceso se ve reflejado en la figura 4:

**Figura 4.**  Visualización del número en base 2

Con el número convertido en base 2, es posible realizar el formato de coma flotante; previamente se ha calculado el signo, ahora es menester calcular el exponente, como se puede observar en la figura 5:

**Figura 5.**  Cálculo del exponente

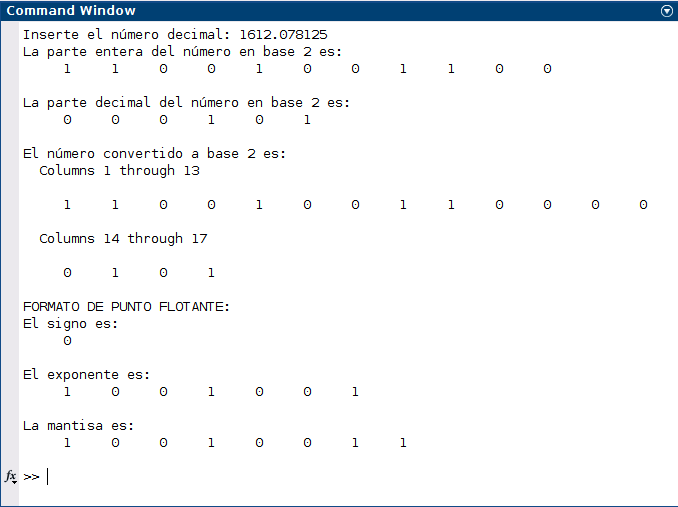
Finalmente, se calcula la mantisa, como se puede apreciar en la figura 6:

**Figura 6.**  Cálculo de la mantisa

1. Usar los siguientes números decimales, para probar la función creada:

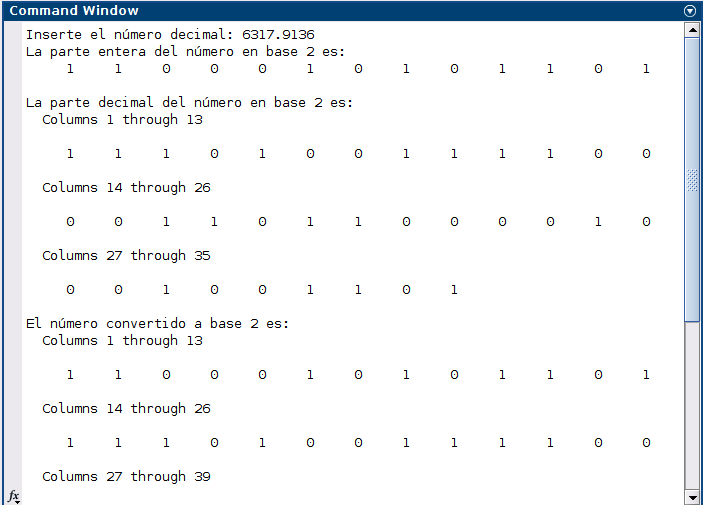
* 1612.07812510

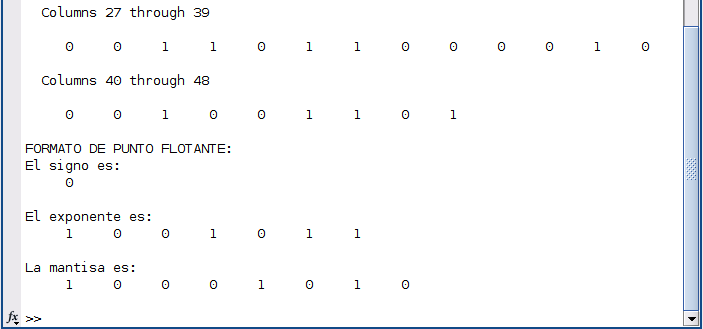
El resultado que muestra el algoritmo desarrollado se puede observar en la figura 7:

**Figura 7.**  Formato de coma flotante de 1612.07812510

* 6317.913610

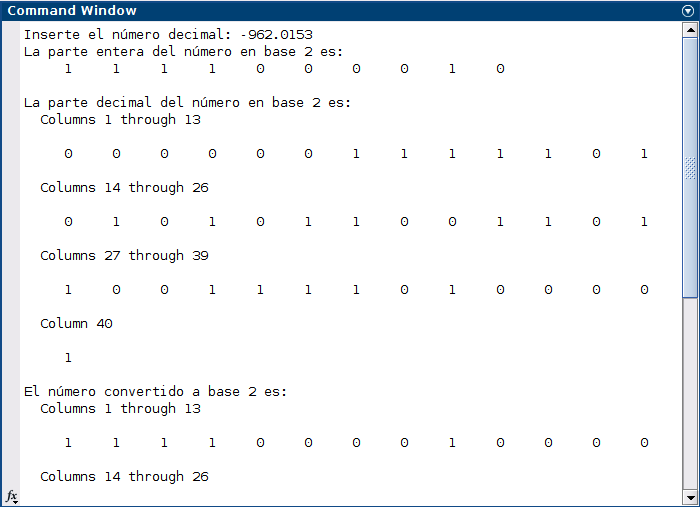
El resultado que muestra el algoritmo desarrollado se puede observar en las figuras 8 y 9:

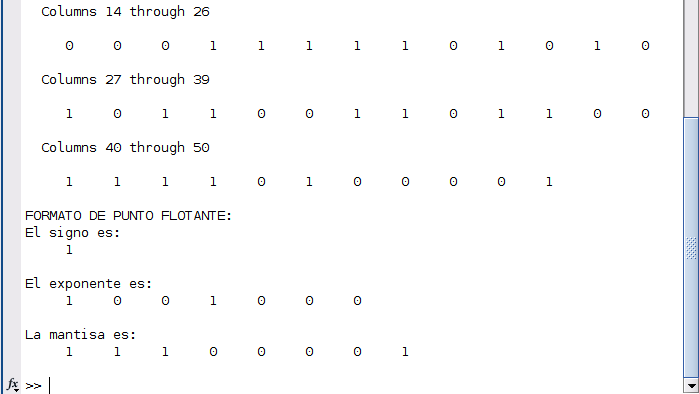
**Figura 8.**  Formato de coma flotante de 6317.913610 (parte 1).

**Figura 9.**  Formato de coma flotante de 6317.913610 (parte 2).

* −962.015310

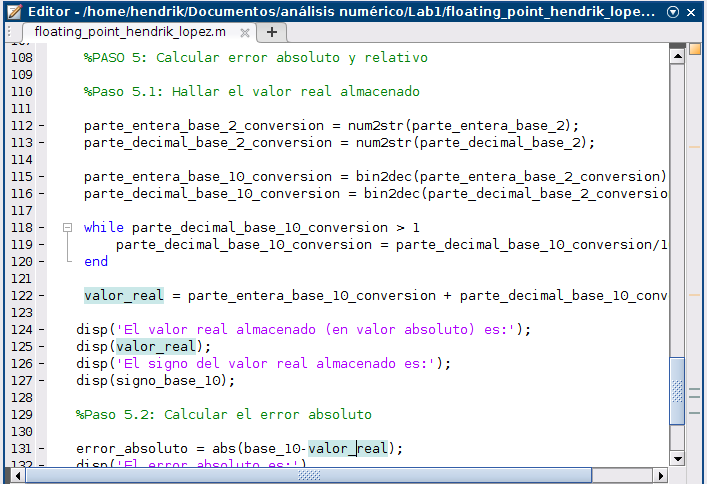
El resultado que muestra el algoritmo desarrollado se puede observar en las figuras 10 y 11:

**Figura 10.**  Formato de coma flotante de -962.015310 (parte 1).

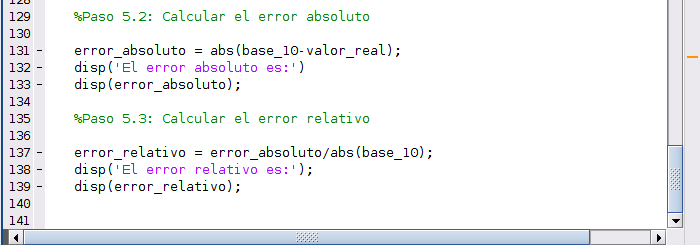
**Figura 11.**  Formato de coma flotante de -962.015310 (parte 2).

1. Calcule el error absoluto y el error relativo entre los números decimales dados y los números almacenados en el computador con 16 bits de precisión.

Para calcular el error absoluto y error relativo, es necesario convertir el valor base 2 almacenado por el computador, a base 10, tal como se puede observar el la figura 12:

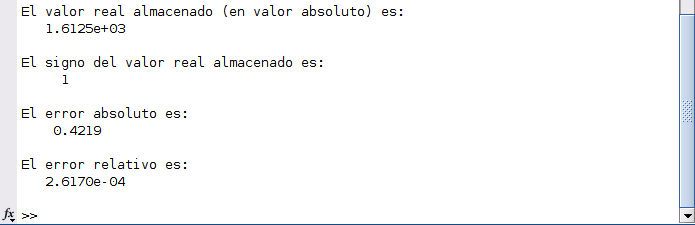
**Figura 12.**  Obtención del valor real.

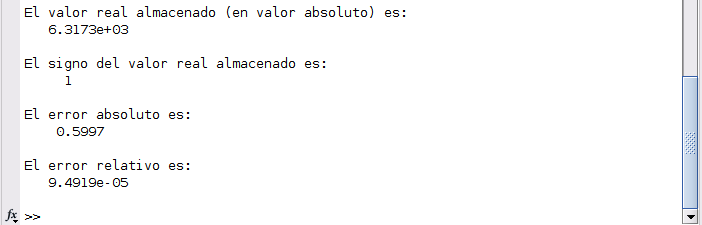
Una vez se ha obtenido el valor real almacenado del número decimal en base 10, es posible calcular el error absoluto y error relativo; esto se puede ver en la figura 13:

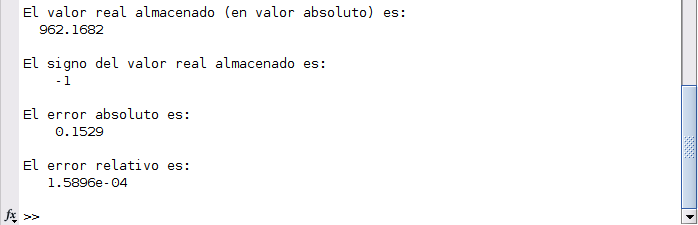


**Figura 13.**  Cálculo del error absoluto y error relativo.

Para probar la eficacia de este algoritmo, las figuras 14, 15 y 16 muestran los resultados, usando los números decimales convertidos en el inciso b; es menester aclarar que, en el caso de los números en base 10, el signo es representado con el número 1; para los negativos, el número correspondiente es -1.

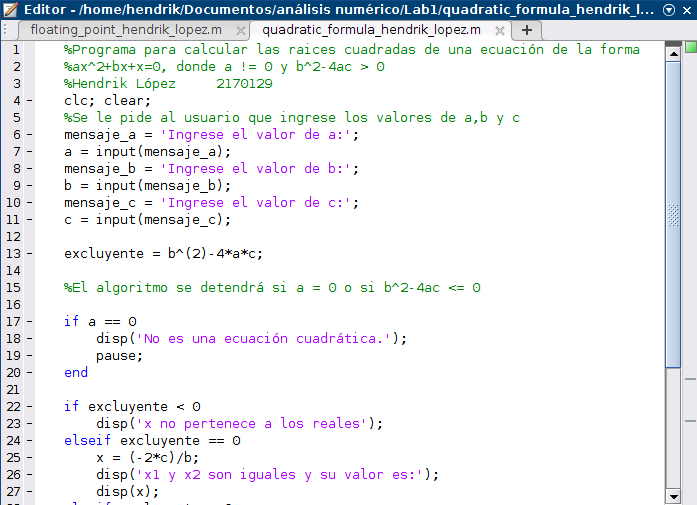
**Figura 14.**  Valor real almacenado de 1612.07812510

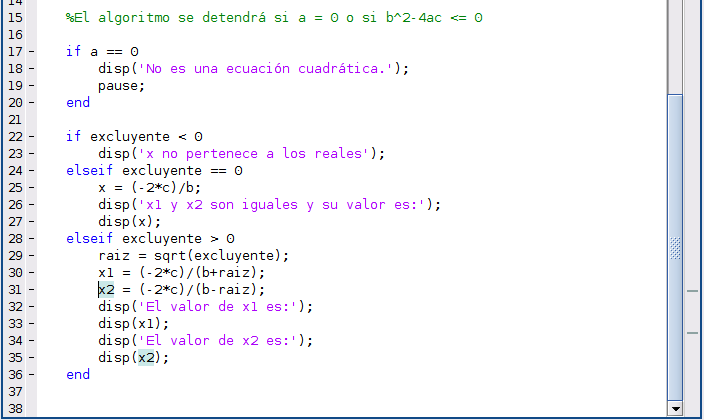
**Figura 15.**  Valor real almacenado de 6317.913610.

**Figura 16.** Valor real almacenado de -962.015310 .

1. Use los resultados obtenidos en el ejercicio *mejorar la ecuación cuadrática* para construir un programa en Matlab que calcule las raíces de la ecuación cuadrática en todas las situaciones, incluyendo el caso donde .

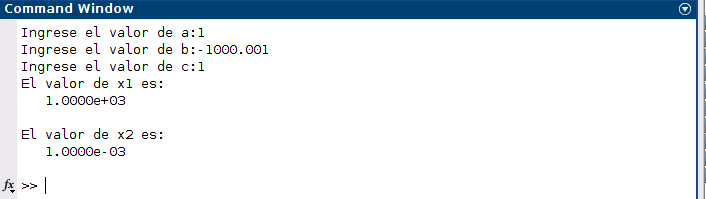
El proceso llevado a cabo consiste en pedirle al usuario que digite los valores de a, b y c; estos son interpretados por el algoritmo y revelan la naturaleza de dicha ecuación. Para ello, detecta si la ecuación no es cuadrática; en ese caso, el programa alerta de ello y se pausa; en caso contrario, analiza el valor del discriminante; si es igual a 0, ambas raíces son iguales; si es mayor a 0, muestra el valor de x1 y x2; si es menor a 0, avisa que la solución de dicha ecuación no se encuentra en los reales. Las figuras 17 y 18 muestran el código implementado en Matlab:

**Figura 17.** Cálculo de raíces de una ecuación cuadrática(parte 1).

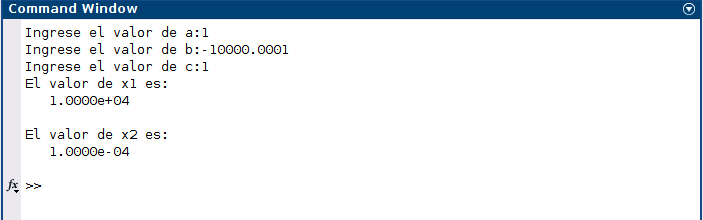
**Figura 18.** Cálculo de raíces de una ecuación cuadrática(parte 2).

1. Encuentre las raíces de las siguientes ecuaciones cuadráticas, usando la función anterior:

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 19:

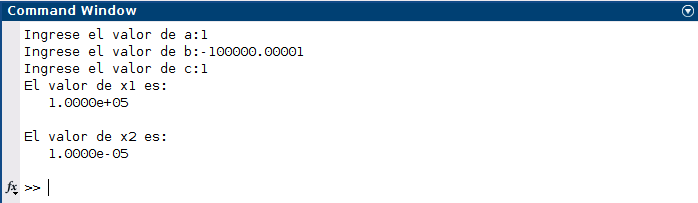
**Figura 19.** Raíces de la ecuación cuadrática .

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 20:



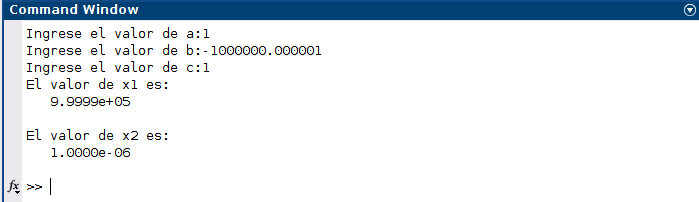
**Figura 20.** Raíces de la ecuación cuadrática

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 21:



**Figura 21.** Raíces de la ecuación cuadrática

El resultado que muestra el algoritmo desarrollado se puede observar en la figura 22:



**Figura 22.** Raíces de la ecuación cuadrática

## Referencias

* Argüello, H. (2021). *Numerical Methods Preliminaries* [Diapositivas]. Universidad Industrial de Santander. https://es.scribd.com/document/423857218/Chapter-1
* Helmenstine, A. (2020, 1 julio). *Determining significant figures*. ThoughtCo. https://www.thoughtco.com/how-to-determine-significant-figures-608326
* The MathWorks, Inc. (2021). *Centro de ayuda*. MathWorks. https://la.mathworks.com/help/index.html