**Calculadora Científica**

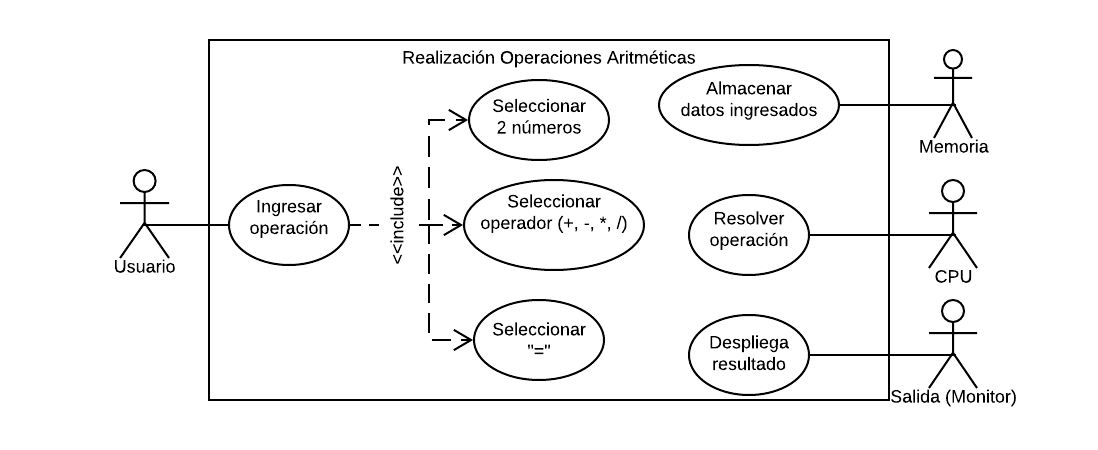
María Fernández Sánchez Cardiel, Nicole Günther Guerrero, Alexander Díaz Ruiz

**Descripción del proyecto**

El proyecto consistió de desarrollar una calculadora capaz de realizar operaciones aritméticas, al igual que sus respectivas funciones trigonométricas, exponenciales, y logarítmicas con tal de poner a prueba nuestros conocimientos y habilidades aprendidos durante el curso.

**Diagramas**

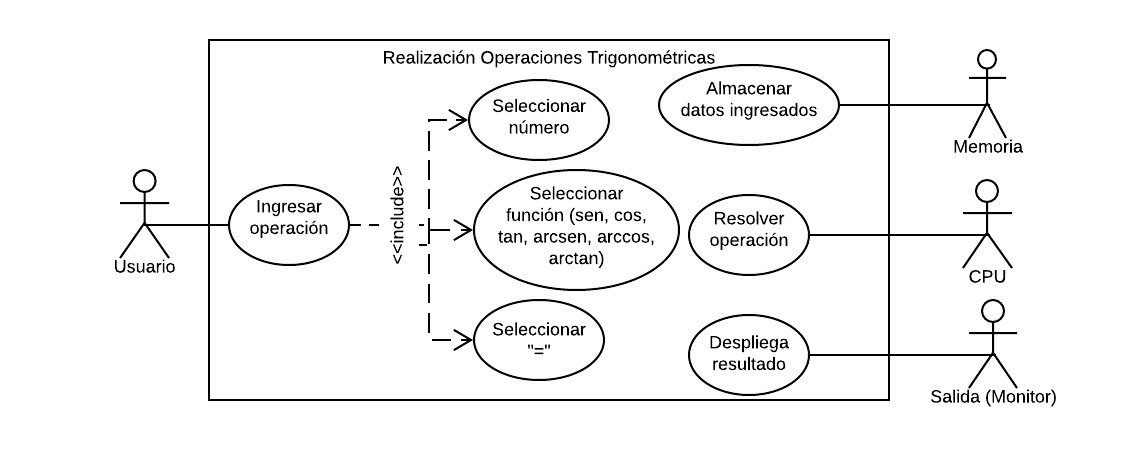
Para facilitar el desarrollo de nuestra aplicación, se emplearon diversos diagramas de tipo Unified Model Language (UML), las cuales sirvieron como modelo a escala del programa a desarrollar:

1. **Casos de Uso:**

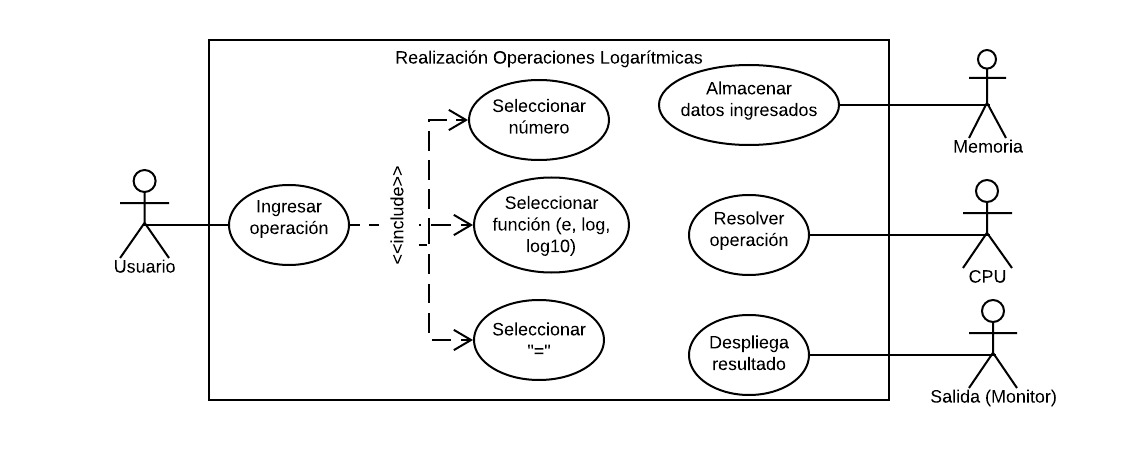
**Figura 1.** Diagrama de *Casos de Us*o para la realización de operaciones aritméticas.

En este *Casos de Uso* vienen presentes cuatro actores; es decir, son cuatro las “entidades” que se involucran directamente con el proceso de las operaciones aritméticas para que éstas se lleven a cabo: el Usuario, la Memoria, el CPU, y la Salida.

El proceso empieza con la participación del Usuario, el cual ingresa la operación a realizar; ésta a su vez se divide en tres subprocesos: el ingreso de dos números—ya sea enteros, flotantes, o *doubles*—, la introducción del tipo de operación que a realizar, y la selección del operador “=”. La Memoria es la encargada de almacenar los datos previamente ingresados; el CPU resolverá la operación con los datos guardados, y la Salida mostrará el resultado a través del monitor.



**Figura 2.** Diagrama de *Casos de Us*o para la realización de operaciones trigonométricas.



**Figura 3.** Diagrama de *Casos de Us*o para la realización de operaciones logarítmicas y exponenciales.

Los siguientes *Casos de Uso* de las **Figuras 2** y **3** son bastante similares a la de la **Figura 1**, ya que todas las operaciones comparten como común denominador los Actores y los Casos de Uso primarios.

Donde diferencian los últimos dos diagramas del primero sería en la cantidad de elementos que se deben ingresar. En las **Figura 2** y **3**, solo se necesita introducir un número para poder realizar la operación; también cabe notar que, en lugar de escoger un operador, el usuario está eligiendo una *función* de la cual se toma el número anterior como parámetro.

1. **Especificaciones de Casos de Uso:**

Las Especificaciones de cada Caso de Uso descrito anteriormente se encuentran a continuación:

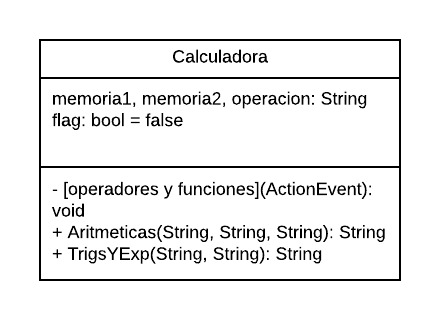
|  |
| --- |
| **Nombre:** Operaciones Aritméticas  **ID:** 114  **Actores:** Usuario, Memoria, CPU, Salida  **Descripción:** Este caso de uso será utilizado para realizar cálculos aritméticos entre dos números utilizando un interfaz.  **Precondiciones:** Tener la calculadora encendida.  **Flujo principal:** introducir primer número, seleccionar operador, introducir segundo número, oprimir botón “=”.  **Postcondiciones:** Resultado.  **Flujos alternativos:** error de operación, sobrecarga de operaciones. |

|  |
| --- |
| **Nombre:** Operaciones Trigonométricas  **ID:** 115  **Actores:** Usuario, Memoria, CPU, Salida  **Descripción:** Este caso de uso será utilizado para realizar cálculos trigonométricos introduciendo sólo un número.  **Precondiciones:** Tener la calculadora encendida.  **Flujo principal:** introducir número, seleccionar operador, oprimir botón “=”.  **Postcondiciones:** Resultado.  **Flujos alternativos:** error de operación, sobrecarga de operaciones. |

|  |
| --- |
| **Nombre:** Operaciones Logarítmicas  **ID:** 116  **Actores:** Usuario, Memoria, CPU, Salida  **Descripción:** Este caso de uso será utilizado para realizar cálculos logarítmicos introduciendo sólo un número.  **Precondiciones:** Tener la calculadora encendida.  **Flujo principal:** introducir número, seleccionar operador, oprimir botón “=”.  **Postcondiciones:** Resultado.  **Flujos alternativos:** error de operación, sobrecarga de operaciones. |

**Figuras 4, 5,** y **6.** Especificaciones de los *Casos de Uso* descritos en las **Figuras 1, 2,** y **3.**

1. **Diagrama de Clase:**



**Figura 7.** Diagrama de *Clase*

El Diagrama de Clases consiste de una sola que se encarga tanto de proyectar la interfaz de la calculadora como de realizar todos los cálculos que el programa ofrece. Entre sus Datos Miembros se encuentran *memoria1*, *memoria2*, y *operacion*—todas de tipo String—, las cuales se encargan de guardar las cadenas de los números y operador/función que la calculadora deberá realizar.

*flag* es un booleano que determina cuándo se ha seleccionado un operador aritmético, con tal de determinar si es necesario borrar el número actualmente escrito en el Text Field de la interfaz para así permitir la proyección de uno nuevo. *flag* nada más es aplicable para operaciones aritméticas, ya que como todas las funciones nada más aceptan un número como argumento, no hay necesidad de guardar dos (y por ende borrando el primero que se haya ingresado en el Text Field).

La clase cuenta con una variedad de métodos responsables del ingreso de cada número, así como selección de operadores, selección de funciones, y su aplicación para calcularlos en sí.

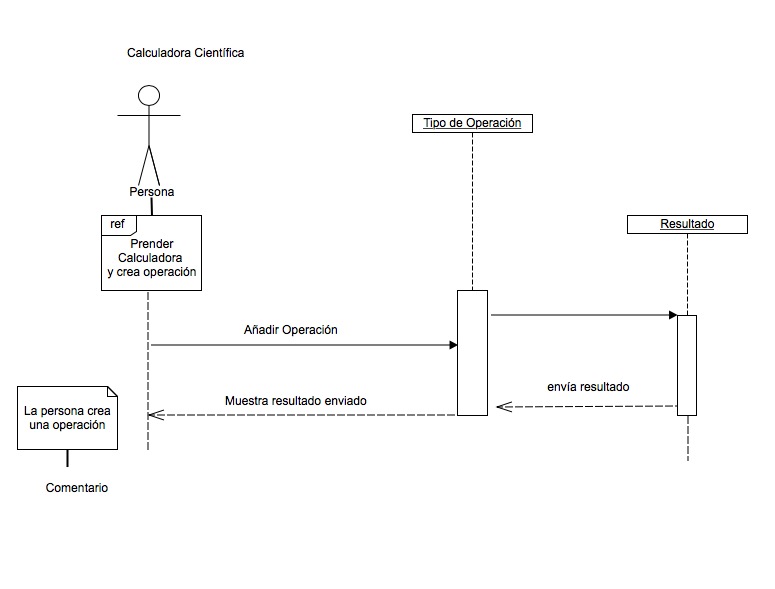
“[operadores y funciones]” engloba todos aquellos métodos responsables de seleccionar los números, operadores, y funciones de la interfaz. Naturalmente, como se utilizan botones para representar éstos, todos aceptan un ActionEvent (AKA *click*) como parámetro para operar. Como ninguna otra clase más que la interfaz puede acceder a estos botones, estos métodos están declarados con un acceso de tipo *público*.

“Aritméticas” y “TrigsYExp” operan de manera muy similar, dado que ambas funciones aceptan Strings como argumentos—el operador/las funciones y el/los número(s) que ocupar en ellos—y devuelven un String nuevo: el resultado de la operación.

El único aspecto que varía entre las dos es el número de parámetros a aceptar. “Aritméticas” acepta tres: 2 números y 1 operador, mientras que “TrigsYExp” sólo requiere de un número para realizar su función.

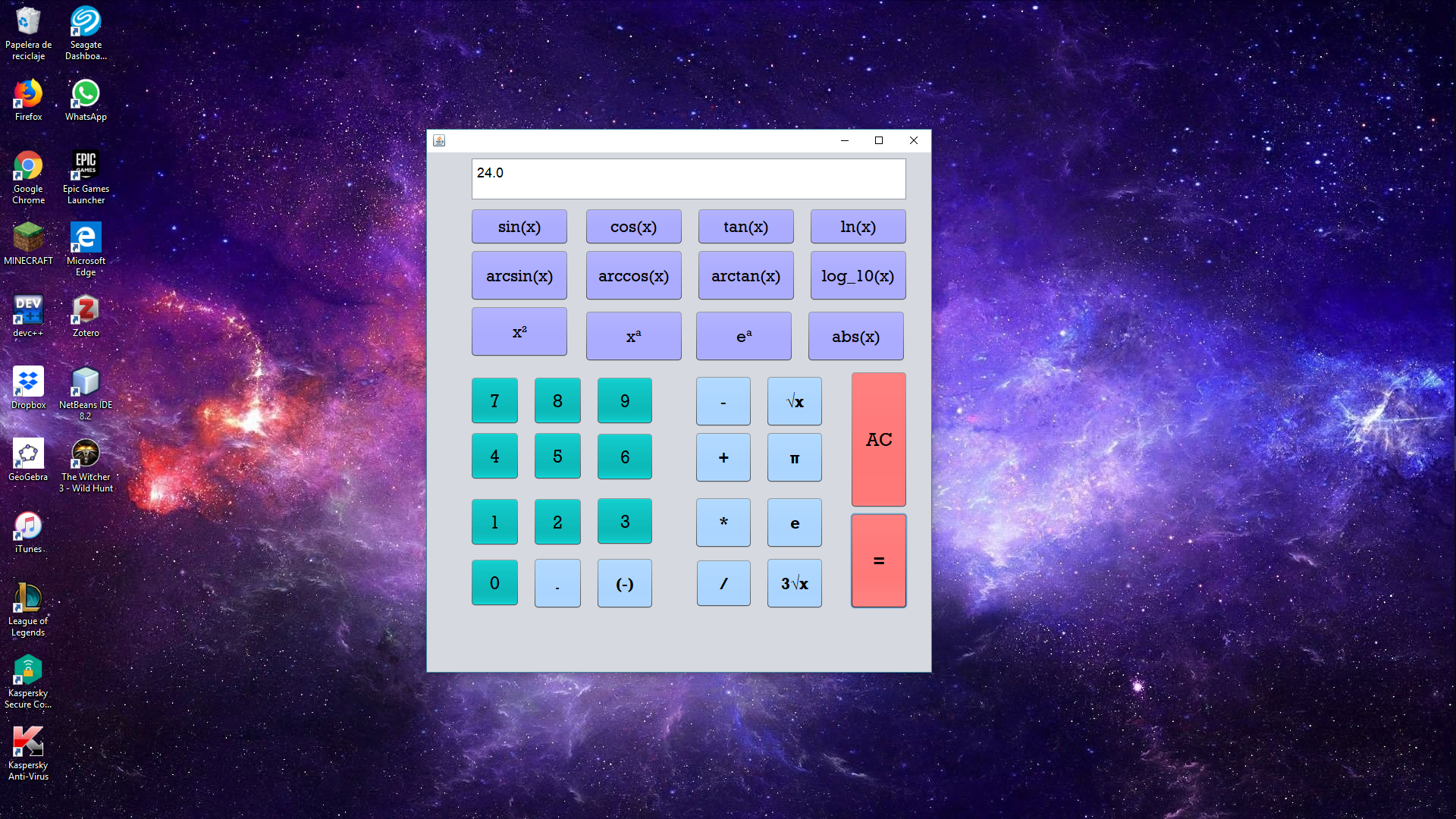
Ambos métodos están declarados con acceso público, refiriéndose a su disponibilidad de ser llamados desde otras clases. Sin embargo, como este programa nada más cuenta con una clase (a pesar de que la posibilidad de haberla hecha en 2 existe: una clase encargada de realizar las operaciones y otra que fuese la interfaz), esta propiedad no se ve en uso.

1. **Diagrama de Secuencia**



**Figura 8.** Diagrama de *Secuencia*

**Prueba:**

****

**Figura 9.** Interfaz gráfica de la calculadora, mostrando el resultado de una operación.

**Reflexión**

Al iniciar este proyecto, no esperábamos tener problema con los diagramas de UML, lo que nos llevó a una no tan grata sorpresa. Al estar acostumbrados a hacer el código desde el principio, sin ningún modelo que nos ayude, y solucionar problemas conforme avanzamos, no aprovechamos el uso de éstos. Los diagramas, en lugar de facilitar el trabajo, nos retrasaron y mostraron ser un reto mayor de lo que estábamos preparados. Este proyecto será una referencia para el futuro; si es necesario utilizar los diagramas, primero empezaremos por ellos y después desarrollaremos el código conforme el modelo propuesto.

Este proyecto también resultó ser un desafío en cómo manejar los tiempos y trabajar en equipo, aprendimos que la comunicación es esencial para mantener el equipo a flote y sin dejar a alguien trabajando más que los demás.