KEVIN ALEXANDER AQUINO VASQUEZ

PROGRAMA CALCULADORA

KODIGO

Tabla de contenido

[Descripcion de directorio 4](#_Toc179139878)

[Clase HandleScanner 5](#_Toc179139879)

[Metodo getInt 5](#_Toc179139880)

[Metodo getDouble 5](#_Toc179139881)

[Clase CustomMessage 7](#_Toc179139882)

[Metodo title 7](#_Toc179139883)

[Metodo out 7](#_Toc179139884)

[Metodo outln 8](#_Toc179139885)

[Clase Numero 9](#_Toc179139886)

[Metodo get 9](#_Toc179139887)

[Metodo set 9](#_Toc179139888)

[Clase Operación 10](#_Toc179139889)

[Metodo suma 10](#_Toc179139890)

[Metodo resta 10](#_Toc179139891)

[Metodo multiplicacion 11](#_Toc179139892)

[Metodo division 11](#_Toc179139893)

[Metodo potencia 12](#_Toc179139894)

[Metodo raiz 12](#_Toc179139895)

[Clase Calculadora 14](#_Toc179139896)

[Metodo suma 14](#_Toc179139897)

[Metodo resta 15](#_Toc179139898)

[Metodo multiplicacion 15](#_Toc179139899)

[Metodo division 15](#_Toc179139900)

[Metodo potencia 16](#_Toc179139901)

[Metodo raiz 16](#_Toc179139902)

[Metodo getParameters 17](#_Toc179139903)

[Clase Main 18](#_Toc179139904)

[Metodo optionsBanner 18](#_Toc179139905)

[Metodo optionsContinue 19](#_Toc179139906)

[Metodo runCalculator 19](#_Toc179139907)

[Metodo main 20](#_Toc179139908)

[Ejecucion del programa 21](#_Toc179139909)

[Menu de opciones 21](#_Toc179139910)

[Suma 22](#_Toc179139911)

[Resta 23](#_Toc179139912)

[Multiplicacion 24](#_Toc179139913)

[Division 25](#_Toc179139914)

[Potencia 26](#_Toc179139915)

[Raiz 27](#_Toc179139916)

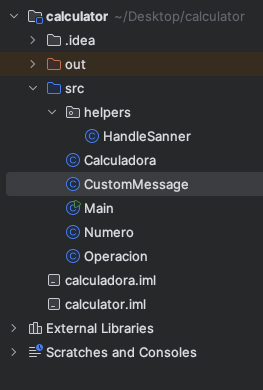
[Continuar 28](#_Toc179139917)

[Finalizar 28](#_Toc179139918)

# Descripcion de directorio

Como se puede visualizar en la ilustracion 1, tenemos nuestro directorio de archivos, los cuales contiene lo siguiente:

* Clase operación
* Clase Numero
* Clase CustomMessage
* Clase Calculadora
* Clase handleScanner
* Clase Main



# Clase HandleScanner

Clase que implementa el uso del Scanner con su respectivas validaciones

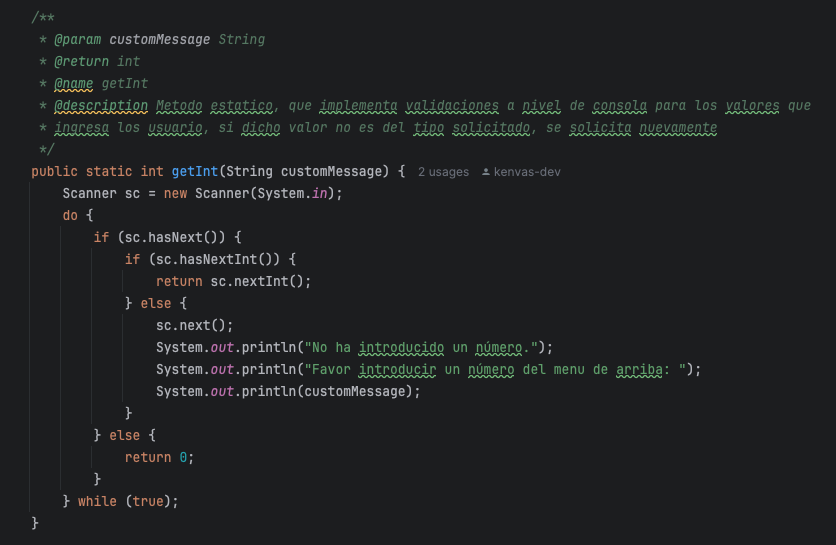
Dicha clase cuenta con dos metodos estaticos, los cuales son:

* getInt
* getDouble

los dos metodos reciben como parametro “customMessage” de tipo String.

## Metodo getInt

Metodo estatico, que implementa validaciones a nivel de consola para los valores que ingresa los usuario, si dicho valor no es del tipo solicitado, se solicita nuevamente



## Metodo getDouble

Metodo estatico, que implementa validaciones a nivel de consola para los valores que ingresa los usuario, si dicho valor no es del tipo solicitado, se solicita nuevamente



# Clase CustomMessage

Clase que implementa el uso del System.out.print.

Clase presonalizada para mostrar mensajes.

Dicha clase cuenta con dos metodos estaticos, los cuales son:

* title
* out
* outln

los dos metodos (out, outln) reciben como parametro “texts” de tipo String[].

El metodo title recibe unicamente el parametro “title” de tipo String

## Metodo title

Banner creado para mostrar enunciados



Ilustración 1 - Metodo title

## Metodo out

Es una personalizacion del syop para un listado de opciones, sin saltos de linea.

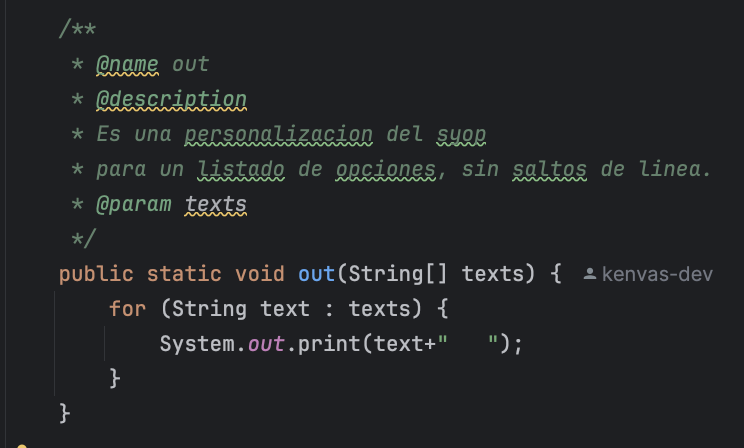


Ilustración 2 - Metodo out

## Metodo outln

Es una personalizacion del syop para un listado de opciones, con saltos de linea.

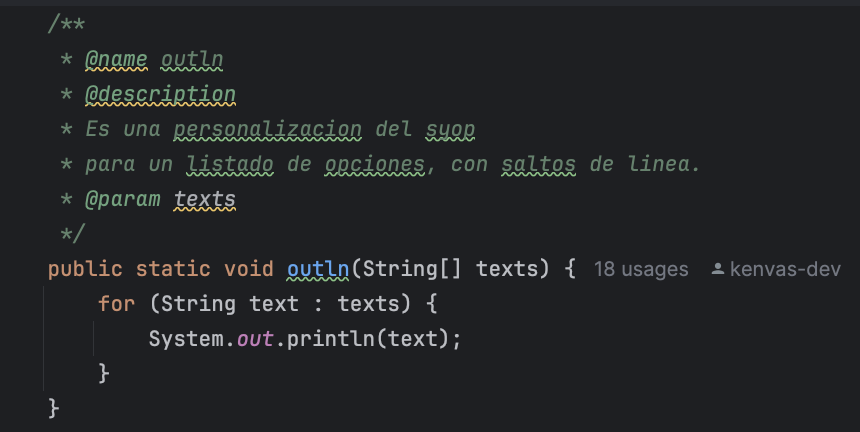


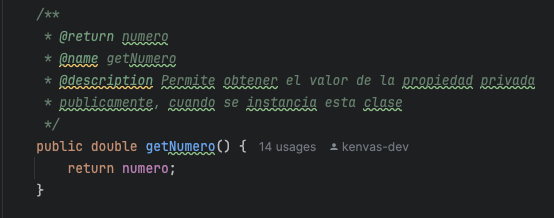
Ilustración 3 - Metodo outln

# Clase Numero

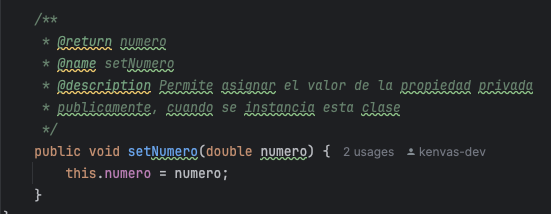
Clase Numero que permite crear un objeto con sus propiedad 'numero'

Consta de una propiedad privada llamada numero de tipo double y sus metodos de acceso, set y get.

## Metodo get



## Metodo set



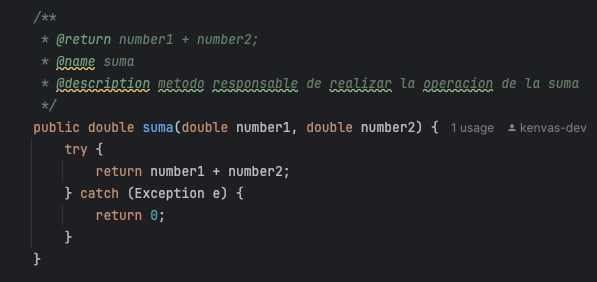
# Clase Operación

En esta clase se realiza la ejecucion de los calculos aritmeticos solicitados, los cuales son:

* SUMA
* RESTA
* MULTIPLICACION
* DIVISION
* POTENCIA
* RAIZ

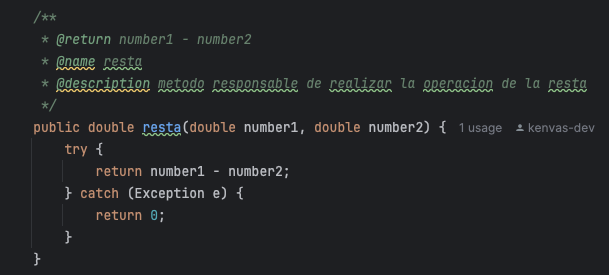
## Metodo suma

Metodo responsable de realizar la operacion de la suma, y retorna el resultado de dicha suma. Si falla retorna ‘0’.



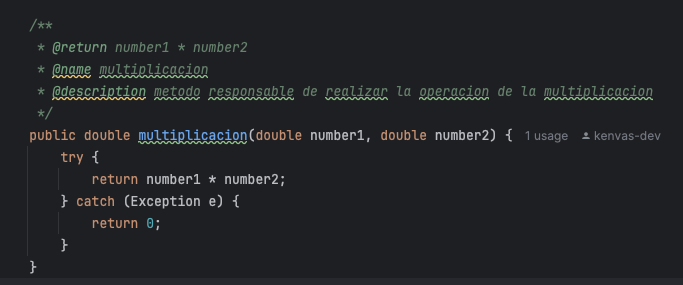
## Metodo resta

Metodo responsable de realizar la operacion de la resta, y retorna el resultado de dicha resta. Si falla retorna ‘0’.



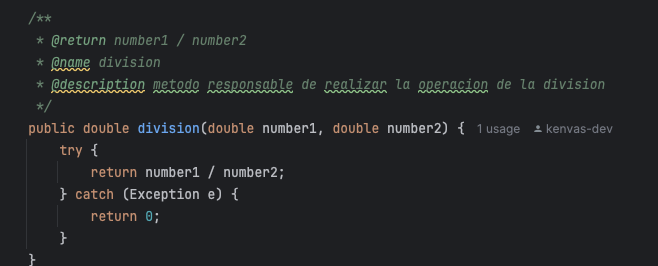
## Metodo multiplicacion

Metodo responsable de realizar la operacion de la multiplicacion, y retorna el resultado de dicha multiplicacion. Si falla retorna ‘0’.



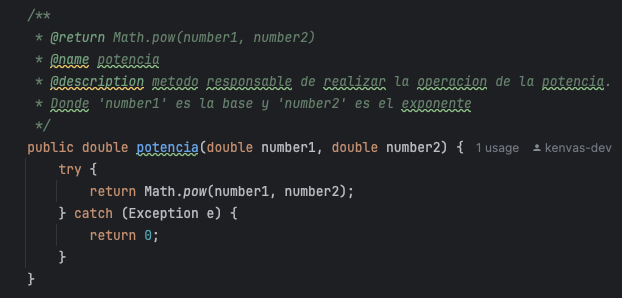
## Metodo division

Metodo responsable de realizar la operacion de la division, y retorna el resultado de dicha division. Si falla retorna ‘0’.



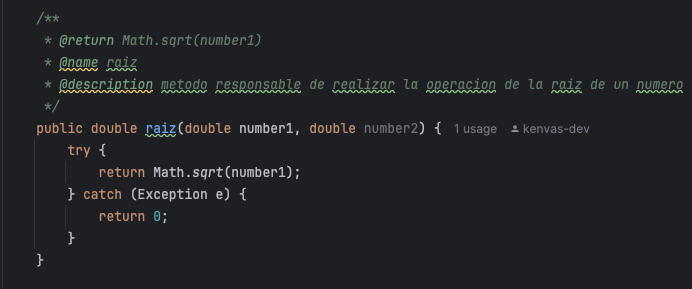
## Metodo potencia

Metodo responsable de realizar la operacion de la potencia, y retorna el resultado de dicha potencia. Si falla retorna ‘0’.



## Metodo raiz

Metodo responsable de realizar la operacion de la raiz, y retorna el resultado de dicha raiz. Si falla retorna ‘0’.



# Clase Calculadora

En esta clase se realiza la implementacion de los calculos aritmeticos solicitados, los cuales son:

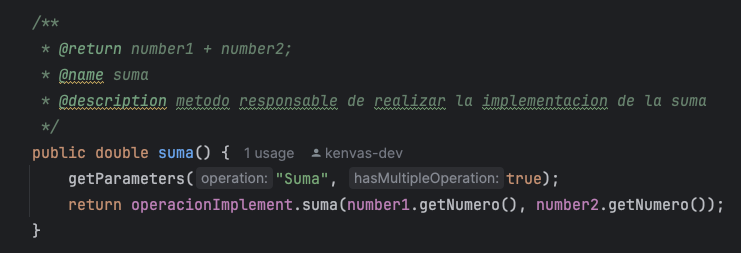
* SUMA
* RESTA
* MULTIPLICACION
* DIVISION
* POTENCIA
* RAIZ

Cuenta con las siguientes propiedades, las cuales son:

* lblNumber1
* lblNumber2
* lblOption
* number1
* number2
* sc
* operacionImplement

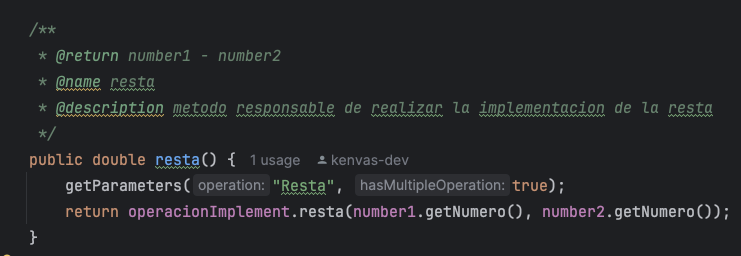
## Metodo suma

Metodo responsable de realizar la implementacion de la suma con sus respectivas validaciones.



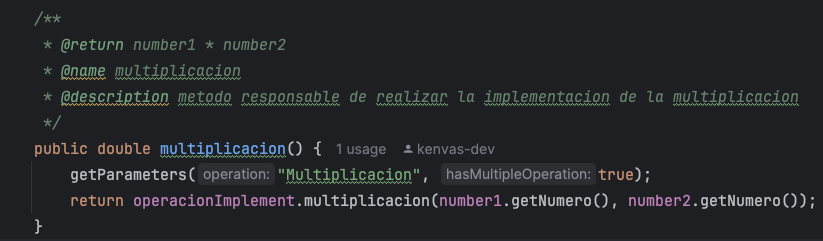
## Metodo resta

Metodo responsable de realizar la implementacion de la resta con sus respectivas validaciones.



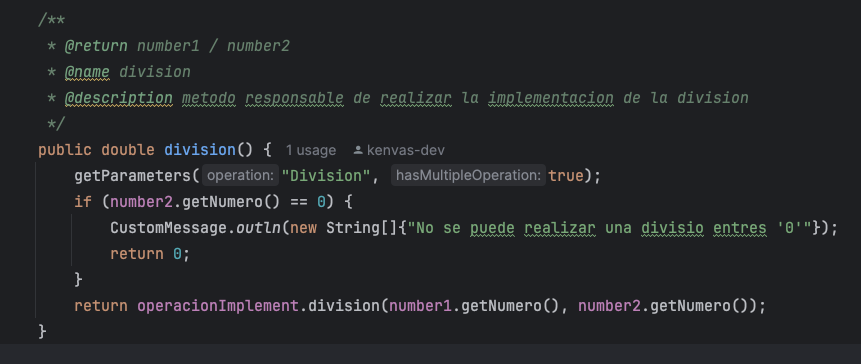
## Metodo multiplicacion

Metodo responsable de realizar la implementacion de la multiplicacion con sus respectivas validaciones.



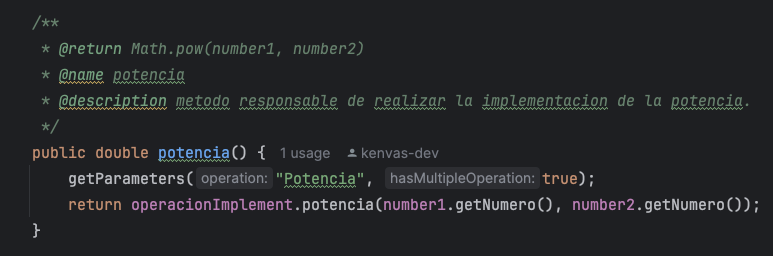
## Metodo division

Metodo responsable de realizar la implementacion de la division con sus respectivas validaciones.



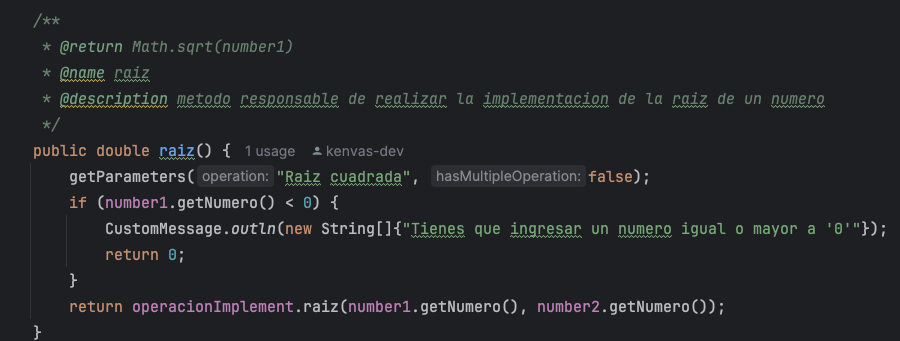
## Metodo potencia

Metodo responsable de realizar la implementacion de la potencia con sus respectivas validaciones.



## Metodo raiz

Metodo responsable de realizar la implementacion de la raiz con sus respectivas validaciones.



## Metodo getParameters

Metodo responsable de solicitar los datos al usuario



# Clase Main

Clase principal la cual llama a la clase calculadora para realizar las operaciones.

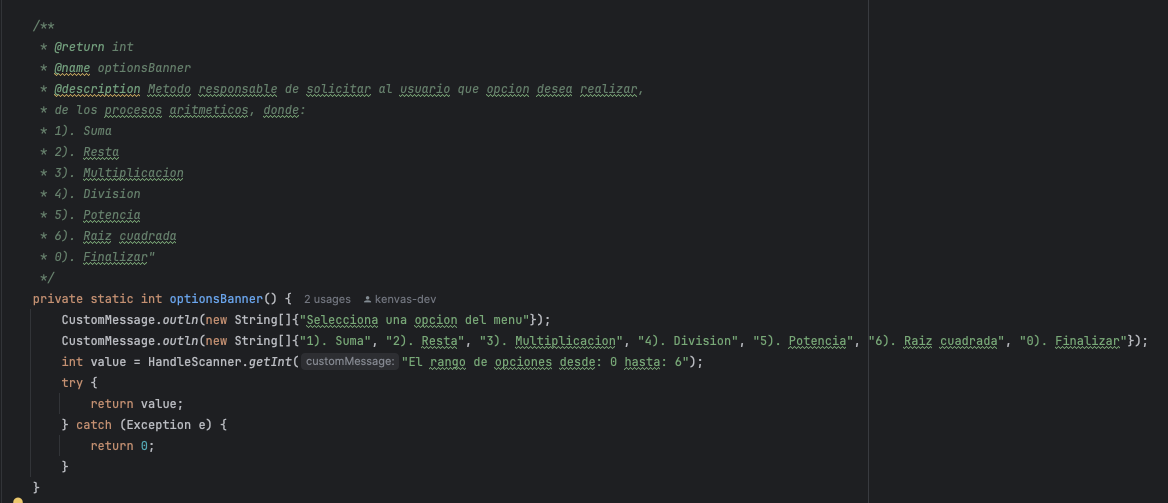
Esta clase cuenta con las siguientes propiedades y metodos:

* minOptionSelected
* optionSelected
* optionSelectedContinue
* hasContinue
* sc
* Metodo main
* Metodo optionsBanner
* Metodo optionsContinue
* Metodo runCalculator

## Metodo optionsBanner

Metodo responsable de solicitar al usuario que opcion desea realizar, de los procesos aritmeticos, donde:

* 1). Suma
* 2). Resta
* 3). Multiplicacion
* 4). Division
* 5). Potencia
* 6). Raiz cuadrada
* 0). Finalizar

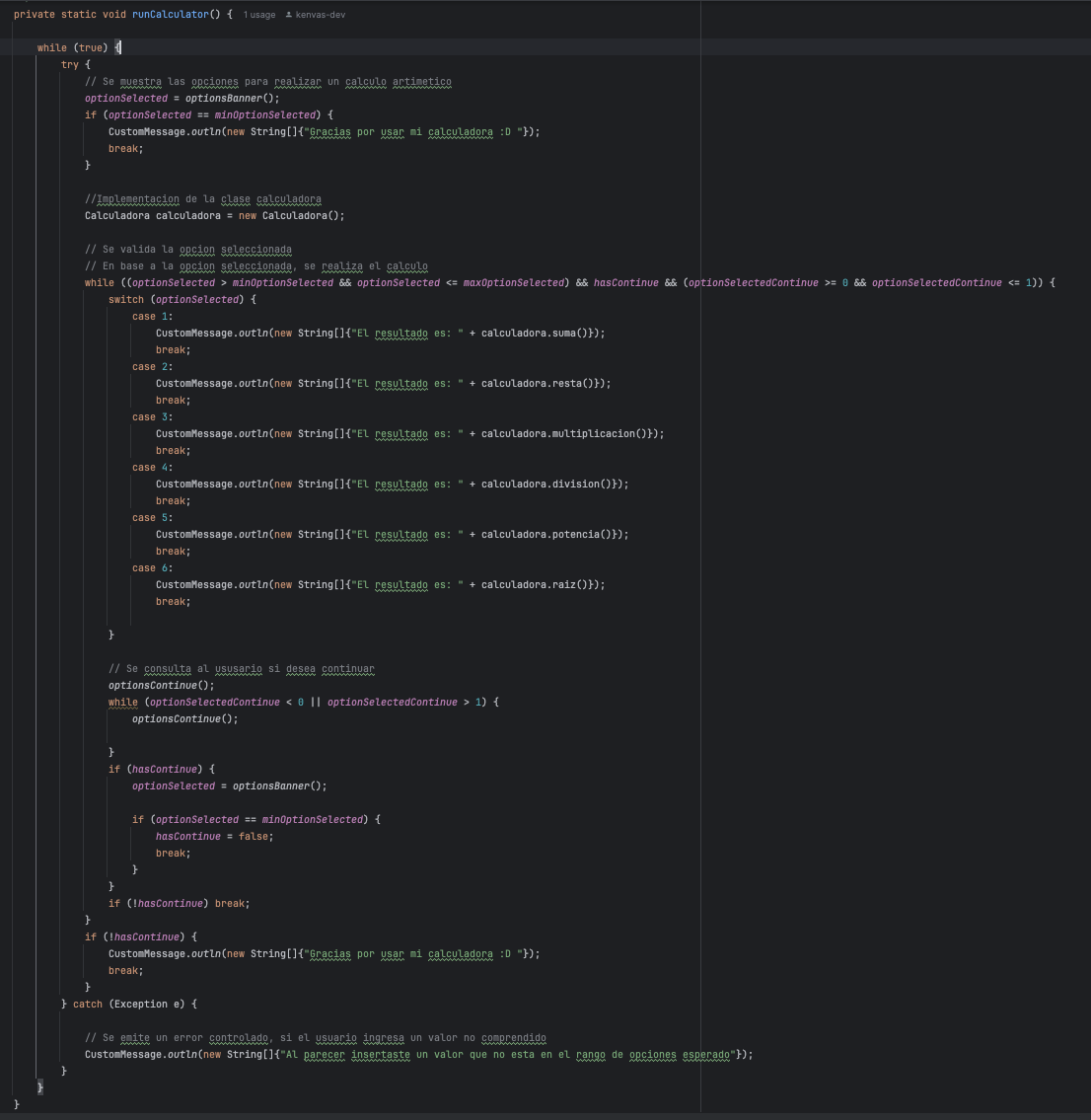


## Metodo optionsContinue

Metodo responsable de solicitar al usuario si desea continuar o no,realizando procesos aritmeticos

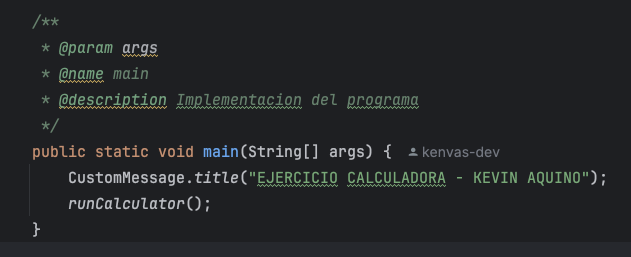
## Metodo runCalculator

Metodo responsable de centralizar los metodos para poder hacer un calculo aritmetico.



## Metodo main

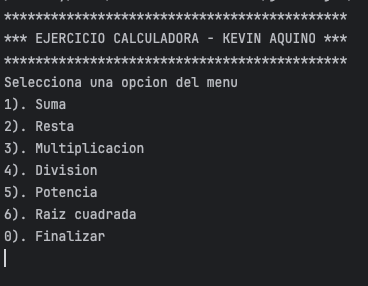
Implementacion del programa



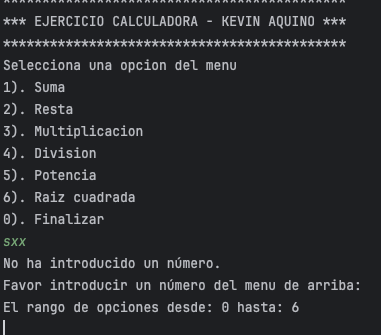
# Ejecucion del programa

## Menu de opciones

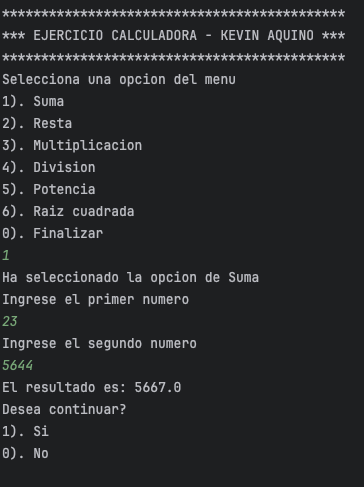
Se debe de digitar una opcion de las 7 que se muestra, si se selecciona una diferente, vuelve a mostrar el mismo mensaje, hasta que ekl usuario digite una correcta, o finalice el programa.



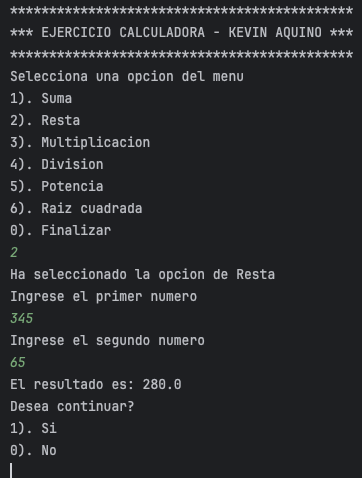
En la siguiente imagen se puede observar si el usuario digita una opcion que no es la correcta, similar es el mensaje cuando se digita algo incorrecto.



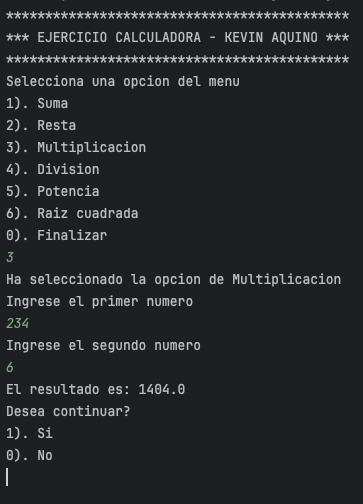
## Suma



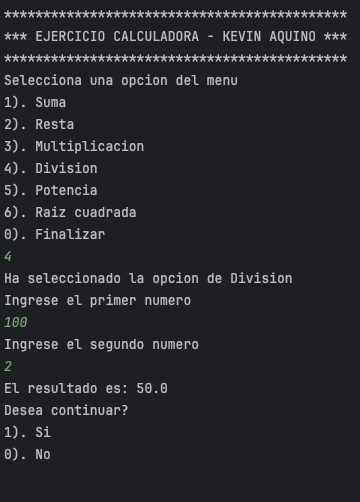
## Resta



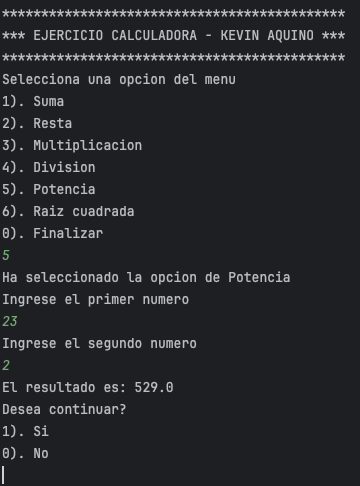
## Multiplicacion



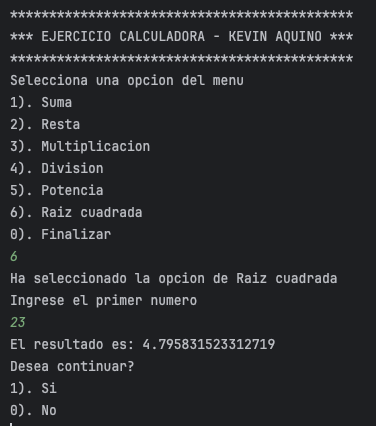
## Division



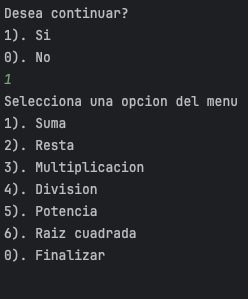
## Potencia



## Raiz



## Continuar



## Finalizar

