**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE**

**Nombre:** Josue Gualotuña

**Asignatura:** Programación Orientada a Objetos

**NRC:** 1939

**Fecha de entrega:** 13 de enero de 2025

**CALCULADORA AVANZADA**

**Descripción del código usando el patrón MVC**

**Modelo**

En esta clase se encuentran los métodos que corresponden a las operaciones que realiza la calculadora, cada método retorna la respectiva operación como un valor de tipo double, además, para las operaciones de raíz cuadrada y potenciación se usa la clase predeterminada Math.

**Vista**

Se encuentran las restricciones o evaluaciones necesarias para que no surjan errores, respecto al ingreso de datos, usando principalmente bucles while, estructuras de control if-else.

Evaluaciones:

* Los valores ingresados deben ser siempre números
* El divisor debe ser siempre diferente de 0
* La raíz cuadrada no aplica en números negativos

**Controlador**

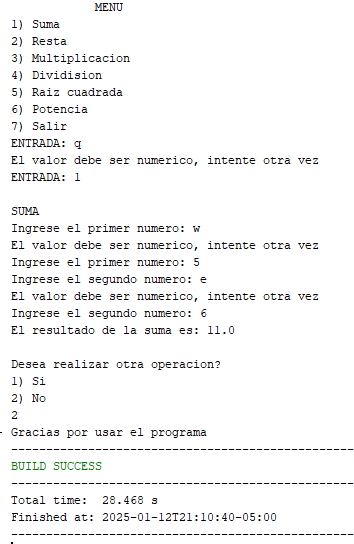
Se realiza un menú con bucle do-while para acceder a las opciones, incluida la de terminar el programa, bucle while para verificar el tipo de dato de la entrada, switch para acceder a cada una de las operaciones y una estructura de control if-else para realizar otra operación o terminar el programa.

**Clase principal**

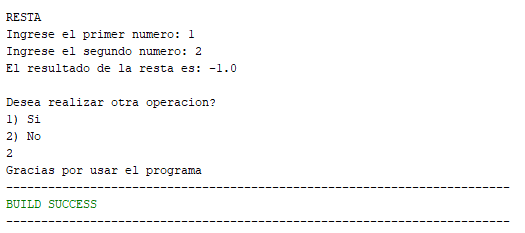
Se inicializa el controlador, teniendo como parámetros instancias de modelo y vista respectivamente.

**Código en funcionamiento**

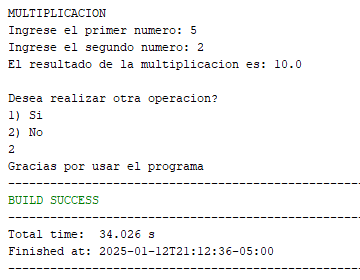
**Menú, suma, comprobación: los valores ingresados deben ser números**

****

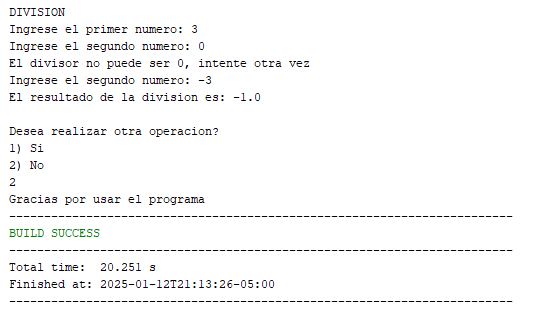
**Resta**

****

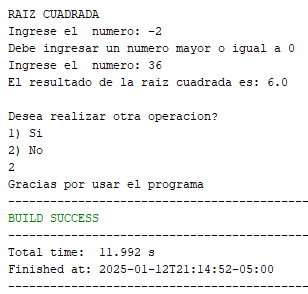
**Multiplicación**

****

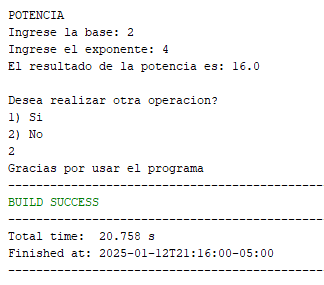
**División, comprobación: el divisor no puede ser 0**

****

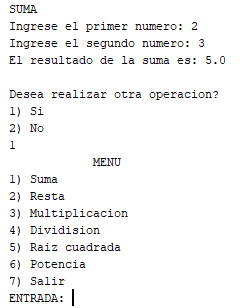
**Raíz cuadrada, comprobación: no se aplica raíz cuadrada en números negativos**



**Potencia**



**Funcionamiento continuo**



**Código del programa:**

**Model**

package model;

public class calculadoraModel {

public double suma(double a , double b ){

return a+b;

}

public double resta(double a, double b){

return a-b;

}

public double multiplicacion(double a, double b){

return a\*b;

}

public double division(double a, double b){

return a/b;

}

public double raiz(double a){

return Math.sqrt(a);

}

public double potencia(double a, double b){

return Math.pow(a, b);

}

}

**View**

package view;

import model.calculadoraModel;

import java.util.Scanner;

public class calculadoraView {

private calculadoraModel modelo;

Scanner entrada = new Scanner(System.in);

public double pedirNumero(String mensaje){

double nueva=0;

while(true){

System.out.print(mensaje);

if(entrada.hasNextDouble()){

nueva=entrada.nextDouble();

break;

}

else{

System.out.println("El valor debe ser numerico, intente otra vez");

entrada.next();

}

}

return nueva;

}

public double pedirDivisor(String mensaje){

double divisor =0;

while(true){

System.out.print(mensaje);

if(entrada.hasNextDouble()){

divisor = entrada.nextDouble();

if(divisor!=0){

break;

}

else{

System.out.println("El divisor no puede ser 0, intente otra vez ");

}

}

else{

System.out.println("El valor debe ser numerico, intente otra vez");

entrada.next();

}

}

return divisor;

}

public double pedirRaiz(String mensaje){

double raiz =0;

while(true){

System.out.print(mensaje);

if(entrada.hasNextDouble()){

raiz= entrada.nextDouble();

if(raiz>=0){

break;

}else{

System.out.println("Debe ingresar un numero mayor o igual a 0");

}

}

else{

System.out.println("El valor debe ser numerico, intente otra vez");

entrada.next();

}

}

return raiz;

}

public void resultado(double resultado){

System.out.println(resultado);

}

}

**Controller**

package controller;

import java.util.Scanner;

import model.calculadoraModel;

import view.calculadoraView;

public class calculadoraController {

private calculadoraModel model;

private calculadoraView view;

public calculadoraController(calculadoraModel model, calculadoraView view) {

this.model = model;

this.view = view;

}

public void iniciar(){

double num1,num2,numResultado;

int opcion=0;

do{

System.out.println(" MENU");

System.out.println("1) Suma");

System.out.println("2) Resta");

System.out.println("3) Multiplicacion");

System.out.println("4) Dividision");

System.out.println("5) Raiz cuadrada");

System.out.println("6) Potencia");

System.out.println("7) Salir");

System.out.print("ENTRADA: ");

Scanner opcionEntrada = new Scanner(System.in);

while(true){

if(opcionEntrada.hasNextInt()){

opcion = opcionEntrada.nextInt();

if(opcion>0 && opcion<8){

break;

}

else{

System.out.println("Debe ingresar un numero valido, intente de nuevo");

System.out.print("ENTRADA: ");

}

}

else{

System.out.println("El valor debe ser numerico, intente otra vez");

System.out.print("ENTRADA: ");

opcionEntrada.next();

}

}

System.out.println();

if(opcion==7){

System.out.println("Gracias por usar el programa");

break;

}

switch(opcion){

case 1:

System.out.println("SUMA");

num1=view.pedirNumero("Ingrese el primer numero: ");

num2=view.pedirNumero("Ingrese el segundo numero: ");

numResultado= model.suma(num1, num2);

System.out.print("El resultado de la suma es: ");

view.resultado(numResultado);

System.out.println();

break;

case 2:

System.out.println("RESTA");

num1=view.pedirNumero("Ingrese el primer numero: ");

num2 = view.pedirNumero("Ingrese el segundo numero: ");

numResultado=model.resta(num1, num2);

System.out.print("El resultado de la resta es: ");

view.resultado(numResultado);

System.out.println();

break;

case 3:

System.out.println("MULTIPLICACION");

num1=view.pedirNumero("Ingrese el primer numero: ");

num2 = view.pedirNumero("Ingrese el segundo numero: ");

numResultado=model.multiplicacion(num1, num2);

System.out.print("El resultado de la multiplicacion es: ");

view.resultado(numResultado);

System.out.println();

break;

case 4:

System.out.println("DIVISION");

num1=view.pedirNumero("Ingrese el primer numero: ");

num2= view.pedirDivisor("Ingrese el segundo numero: ");

numResultado= model.division(num1, num2);

System.out.print("El resultado de la division es: ");

view.resultado(numResultado);

System.out.println();

break;

case 5:

System.out.println("RAIZ CUADRADA");

num1=view.pedirRaiz("Ingrese el numero: ");

numResultado= model.raiz(num1);

System.out.print("El resultado de la raiz cuadrada es: ");

view.resultado(numResultado);

System.out.println();

break;

case 6:

System.out.println("POTENCIA");

num1=view.pedirNumero("Ingrese la base: ");

num2=view.pedirNumero("Ingrese el exponente: ");

numResultado= model.potencia(num1,num2);

System.out.print("El resultado de la potencia es: ");

view.resultado(numResultado);

System.out.println();

break;

}

System.out.println("Desea realizar otra operacion?");

System.out.println("1) Si");

System.out.println("2) No");

while(true){

if(opcionEntrada.hasNextInt()){

opcion= opcionEntrada.nextInt();

if(opcion>0&&opcion<3){

break;

}

else{

System.out.println("Opcion no valida, intente de nuevo");

opcionEntrada.next();

}

}

else{

System.out.println("El valor debe ser numerico, intente de nuevo");

opcionEntrada.next();

}

}

if(opcion==1){

}

else{

System.out.println("Gracias por usar el programa");

break;

}

}

while(opcion!=7);

}

}

**Clase principal**

package prycalculadoraavanzada;

import controller.calculadoraController;

import model.calculadoraModel;

import view.calculadoraView;

public class PryCalculadoraAvanzada {

public static void main(String[] args) {

calculadoraModel model = new calculadoraModel();

calculadoraView view = new calculadoraView();

calculadoraController main = new calculadoraController(model,view);

main.iniciar();

}

}