

Resolución paso a paso

Paso A: Las tareas se distribuyeron equitativamente, asignando dos tareas a cada miembro del grupo. Joaquín Silva se encargó de crear un menú que conectó todos los métodos y realizó la Tarea 3, que implicaba calcular perímetros, áreas y volúmenes según correspondía para las siguientes figuras geométricas: cuadrado, rectángulo, círculo, esfera, cubo y cono. Gerson Urrea asumió la responsabilidad de las Tareas 1 y 5, que consisten en calcular las cuatro operaciones aritméticas principales y calcular la ecuación de una recta. Por último, Sebastián Martínez se encargó de las Tareas 2 y 4, que involucraron calcular las soluciones de una ecuación cuadrática y obtener las soluciones de un sistema de ecuaciones. Finalmente se realizaron pruebas unitarias

Paso B: Después de realizar las pruebas unitarias, es posible identificar diversas excepciones que pueden ser aprovechadas en nuestro beneficio para minimizar errores y aplicar try-catch, contribuyendo así a mejorar nuestro código.

Le ponemos la respuesta a esto:

B.) Ahora a partir de lo anterior, considere lo siguiente:

- ¿Qué pasa si se intenta ingresar como divisor un CERO?

Inicialmente, al intentar utilizar 0 como divisor, se obtenía un resultado de "infinity" o, en otras palabras, una indeterminación. Este problema se resolvió mediante la implementación de un método que previene este escenario.

- ¿Qué pasa si la base y exponente de la potencia son CERO?

En las etapas iniciales del método de cálculo de potencias, cuando se utilizaba 0 como base y exponente, se obtenía un resultado de 1 debido a un diseño preestablecido de la función Math.pow. Para abordar este problema, se implementó un método que evita que ambos valores sean simultáneamente iguales a 0.

- ¿Qué pasa si se intenta ingresar una variable de entrada no numérica?

Al intentar ingresar una variable no numérica, se producía un error de InputMismatchException. Para resolver este problema, se implementó un bloque try-catch en el cual la sección catch maneja esta excepción al mostrar un mensaje de error correspondiente.

Paso C: A partir de todas las soluciones desarrolladas, se subieron todos los códigos a GitHub con el propósito de llevar a cabo un merge que integrara todas estas soluciones en un único conjunto de código.