

# Experiencia Gamificada de pruebas

Walfred Pérez

## Reto 1

Analizarlo, compilarlo y ejecutarlo, validando que no haya errores de sintaxis ni semántica.

### ¿Qué solución presenta el código?

Una calculadora que pide dos valores y realiza la operación de suma, resta, multiplicación y división hasta que el usuario eligen la opción de no continuar.

### ¿Cómo comprobaron que no había errores de sintaxis?

Compila y se ejecuta desde la primera vez

### ¿Cómo comprobaron que no había errores semánticos?

Sí existen errores semánticos, pues ciertos valores puede producir errores, tal es el caso de la división, ya que si se ingresa el número 0 en el valor2 para la opción división, el programa se interrumpe ya que la división entre 0 es indefinida y no se agregó esa validación.

## Reto 2

### ¿Qué hace la prueba unitaria?

Son pruebas para probar partes de la solución. Aplicando el enfoque bottom-up, se comprueban las unidades más pequeñas de funcionalidad antes de moverse a piezas más grandes.

### ¿Cuántas pruebas realizaron para demostrar que funciona bien?

Se realizó solo el reto 1

## Reto 3

### ¿Para qué sirven las excepciones? ¿Cuándo se ejecuta finally?

Para capturar errores potenciales, las excepciones son detecciones de error durante la ejecución de un programa.

Finally siempre se ejecuta, independientemente de si se inicia una excepción.

### ¿Qué estrategias defensivas usaron y por qué?

Se realizó solo el reto 1, pero se podría utilizar por ejemplo la agregación de código que verifique si el valor2 ingresado sea diferente de 0 para poder realizar la división.

### **¿Cómo convertirían el código en clases para poderlo reutilizar?**

La calculadora como tal podría ser una clase.

### **Reto 4**

#### **¿Qué es un caso de pruebas?**

Un caso de prueba se caracteriza por tener una entrada conocida y una salida esperada, los cuales son formulados antes de que se ejecute la prueba.

#### **¿Lo usuarios validaron o verificaron? ¿Por qué?**

Los usuarios validan, porque ellos comprueban el producto. Como no tuvimos usuarios que no fueran los catedráticos compañeros del grupo, en este caso nosotros como usuarios lo validamos y desde el punto de vista de programadores lo verificamos.

#### **¿Qué hubiera pasado si la validación no acepta el sistema? Dé sus recomendaciones**

Si el sistema falla las pruebas de aceptación es porque el problema original no fue comprendido y la solución resultante no lo representa.

Como recomendación se debe comprender el problema original y después de la solución realizar las pruebas de validación ya sea de manera automática o manual. Recordar que para los usuarios la manual es más importante para ellos.