**Taller: Pruebas y Estrategias**

**Junit**

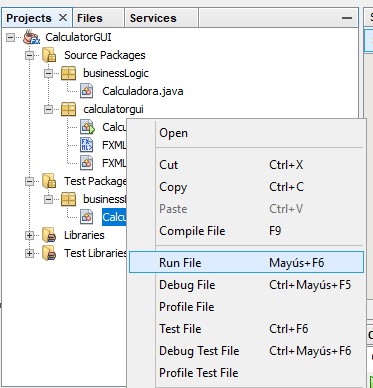
JUnit es un Framework de Pruebas Unitarias de código abierto para JAVA. Es útil para los desarrolladores de Java escribir y ejecutar pruebas repetibles. Fue desarrollado inicialmente por Erich Gamma y Kent Beck. Es una instancia de la arquitectura xUnit y como su nombre lo indica, se utiliza para pruebas unitarias de una pequeña parte del código.

**Calculadora**

La clase calculadora cuenta con las funciones de suma, resta, multiplicación y división. Esta clase está programa con cuatro métodos para realizar las operaciones básicas, y están programados de la forma más sencilla posible. Las funcionalidades de: Atrás, Modulo, Signo, Coma, 10 elevado a una potencia, Raíz Cuadrada, Factorial y Logaritmo deben ser implementados por usted.

Por cada método público de la clase a ser probado se debe crear un método de prueba unitaria de la forma *test\_nombredeOperacion ().* Cada método debe incluir Assert.*assertEquals()* y es necesario remover los llamados a *fail* pues sino todas pruebas van a fallar.

Para ejecutar todos los métodos de pruebas, seleccione clic derecho en el archivo *CalculadoraIT* y seleccione Run File, como se muestra en la Figura 1:



**Figura 1: Ejecutar todos los métodos de prueba**

Todas las pruebas deben pasar satisfactoriamente y se puede observar en la Figura 9, el resultado de la ejecución de las pruebas.

**Orientaciones del taller**

1. Se van a repartir dos funcionalidades por grupo. Cada grupo deberá pensar todas las maneras necesarias para probar la correcta funcionalidad de los métodos.
2. Es posible se realicen cambios al código para asegurar que todas las pruebas pasen.
3. Las pruebas deben ser diseñadas y consignadas en el artefacto de Casos de Prueba.

**Casos de prueba**

A continuación, se muestran un conjunto de casos de pruebas para el ejemplo utilizado en este taller. Se utiliza un formato aleatorio, el equipo de pruebas puede decidir utilizar otro formato más específico si lo desea.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No. Caso**  **Prueba** | **Descripción** | **Juego**  **de**  **Datos** | **Resultados Esperados** | **Resultados**  **Reales** | **Responsable** |
| CP01 | Probando método sumar dos números:  twoAndThreeIsFive | a=2  b=5 | Valor=5 | Passed |  |
| CP02 | Probando método sumar dos números:  twoAndZeroIsTwo | a=2  b=0 | Valor=2 | Passed |  |
| CP03 | Probando método sumar dos números: twoAndMinusTwoIsZero | a=2  b=-2 | Valor=0 | Passed |  |
| CP04 | Probando método sustraer dos  números:  threeMinusTwoIsOne | a=3  b=2 | Valor=1 | Passed |  |
| CP05 | Probando método sustraer dos  números:  threeMinusThreeIsZero | a=3  b=3 | Valor=0 | Passed |  |
| CP06 | Probando método sustraer dos  números:  threeMinusMinusThreeIsSix | a=3  b=-3 | Valor=6 | Passed |  |
| CP07 | Probando método multiplicar  dos números:  ThreeXThreeIsNine | a=3  b=3 | Valor=9 | Passed |  |
| CP08 | Probando método multiplicar  dos números:  threeXZeroIsZero | a=3  b=0 | Valor=0 | Passed |  |
| CP09 | Probando método multiplicar  dos números:  threeXMinusThreeIsMinusNine | a=3  b=-3 | Valor=-9 | Passed |  |
| CP10 | Probando método dividir dos  números:  eigthBetweenTwoIsFour | a=8  b=2 | Valor=4 | Passed |  |
| CP11 | Probando método dividir dos  números: FiveBetweenZeroIsExeption | a=5  b=0 | Valor=AritmethicException  Mensaje: No se puede dividir entre Cero | Passed |  |
| CP12 | Probando método módulos dos números:  as | a=12  b=5 | Valor =2 | Passed |  |
| CP13 | Probando método módulos dos números: | a=2  b=3 | Valor =2 | Passed |  |
| CP14 | Probando ax10^b | a=5  b=3 | Valor =5000 | Passed |  |
| CP15 | Probando raíz cuadrada de un numero | a=9 | Valor =3 | Passed |  |
| CP16 | Probando raíz cuadrada de un numero | a=452.321 | Valor = 672.54816 | Passed |  |
| CP17 | Probando raíz cuadrada de un numero | a=-3 | Valor=AritmethicException  Mensaje: No se puede | Passed |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |