

Entornos de Desarrollo
Unidad 10 - Mi primer TDD V2.0

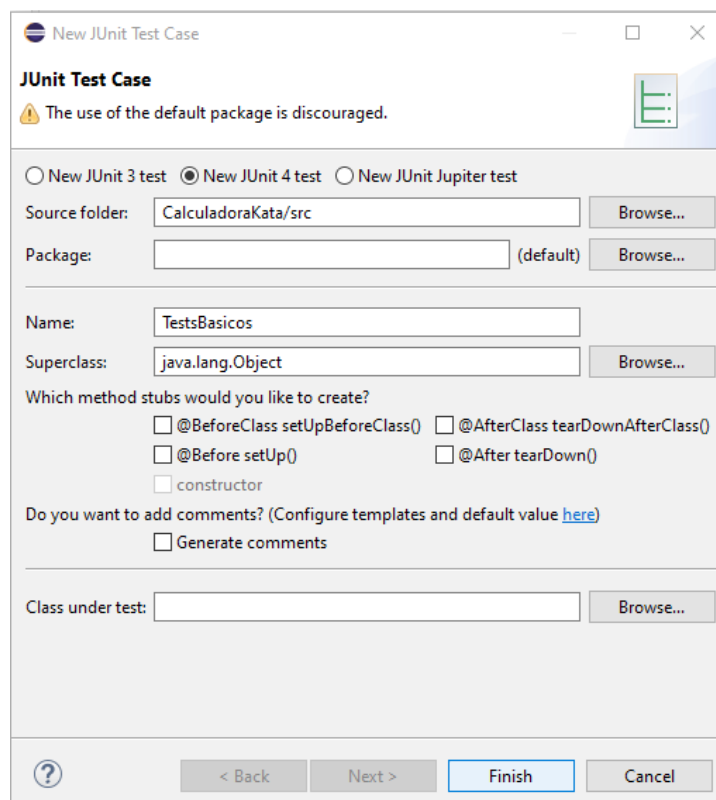
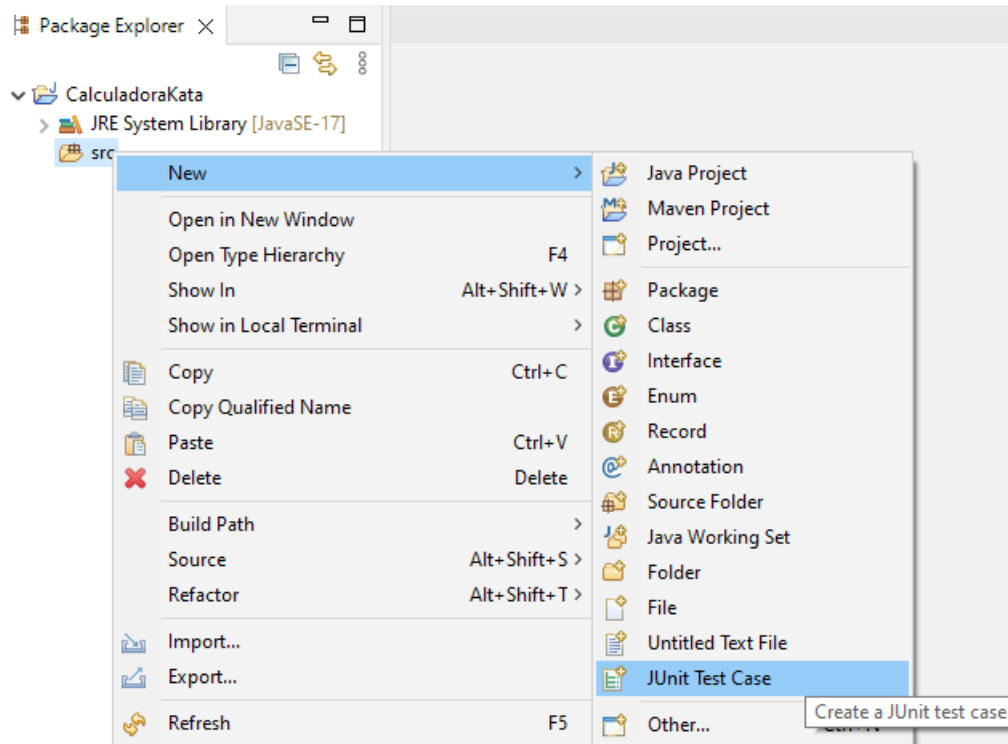
Carlos Blasco Espada
1º Q - DAM

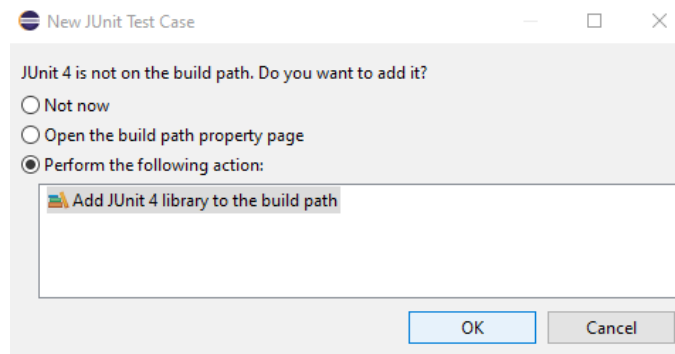
Pasos previos:

- Necesitamos el entorno de desarrollo Eclipse.
- Crear el proyecto.

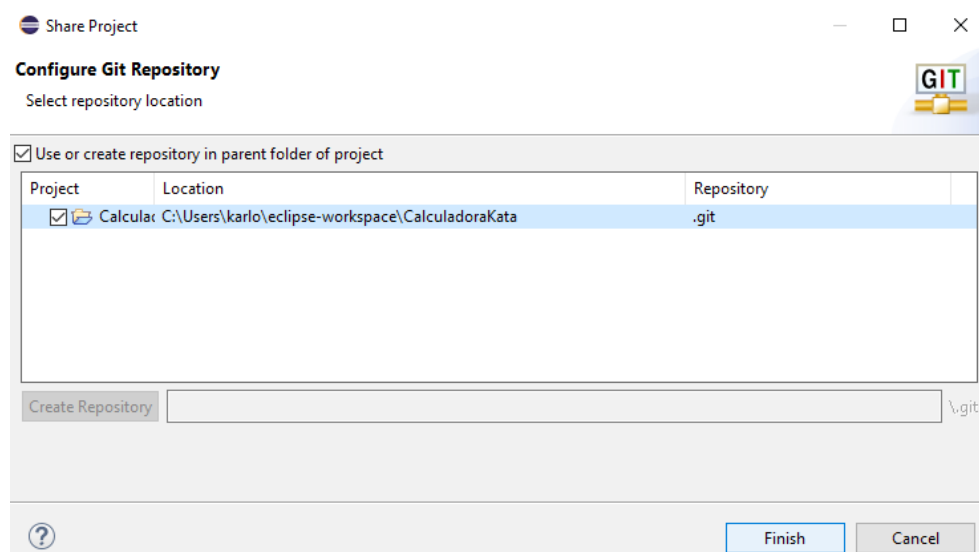
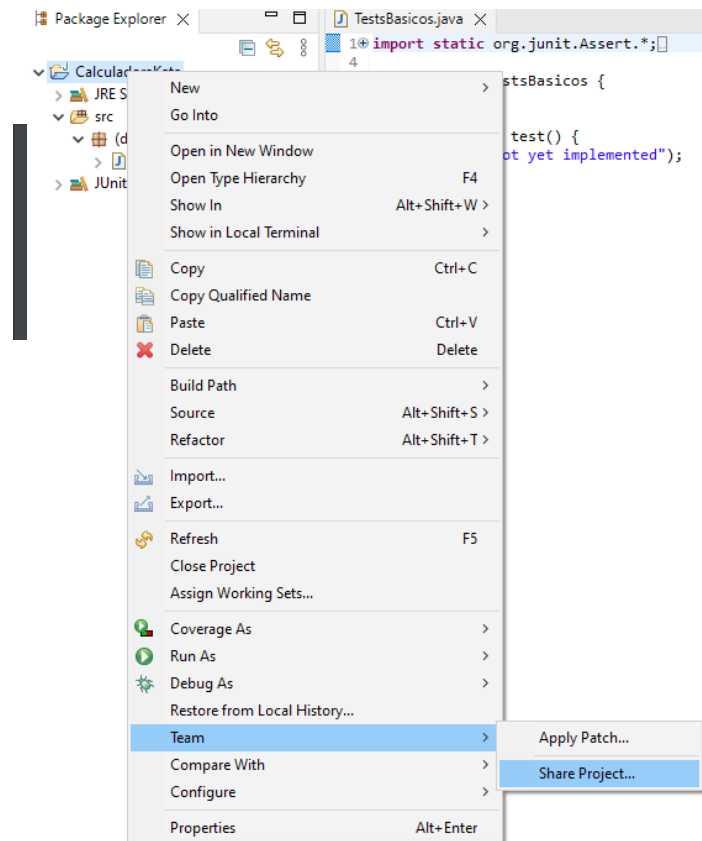
Pasos a seguir:

1. Creamos un nuevo test. Seleccionamos New JUnit 4 test.

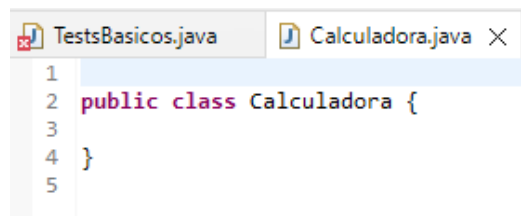
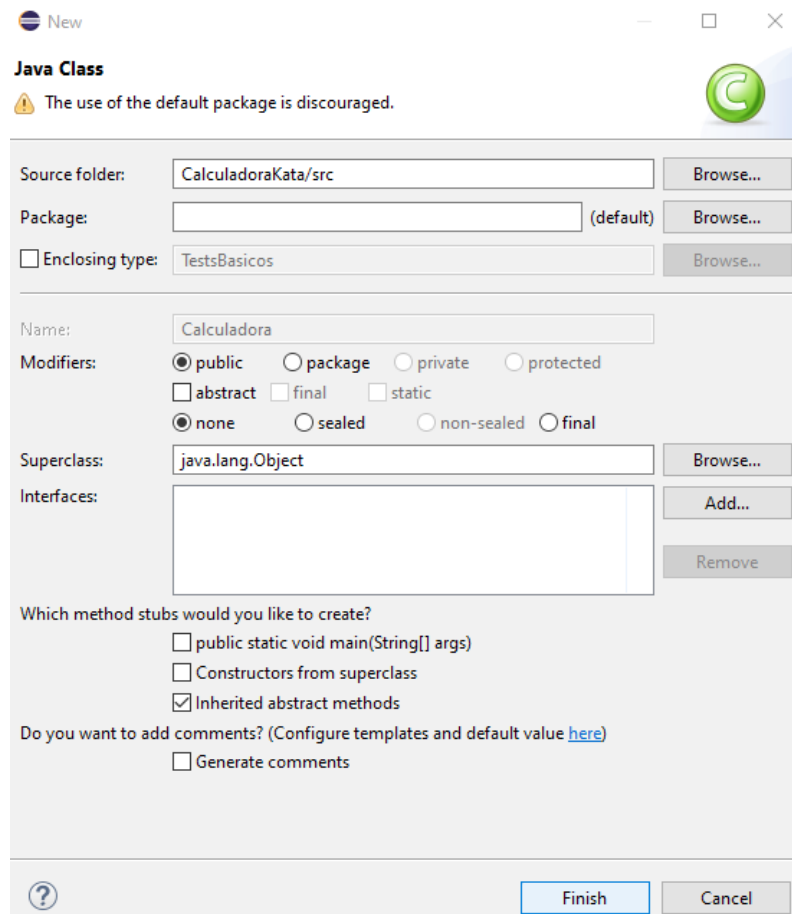
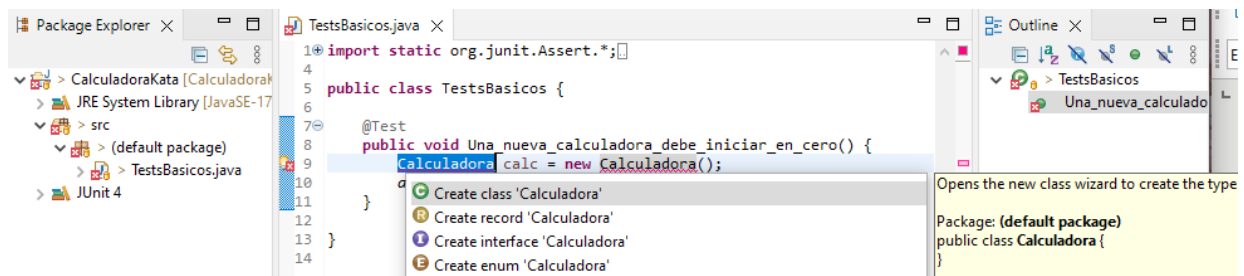




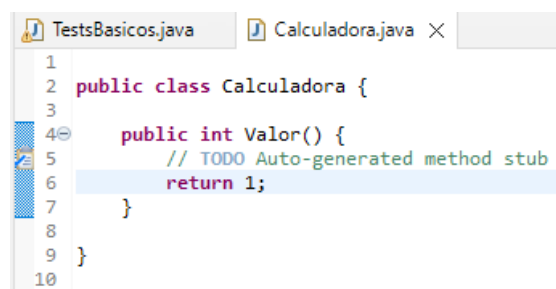
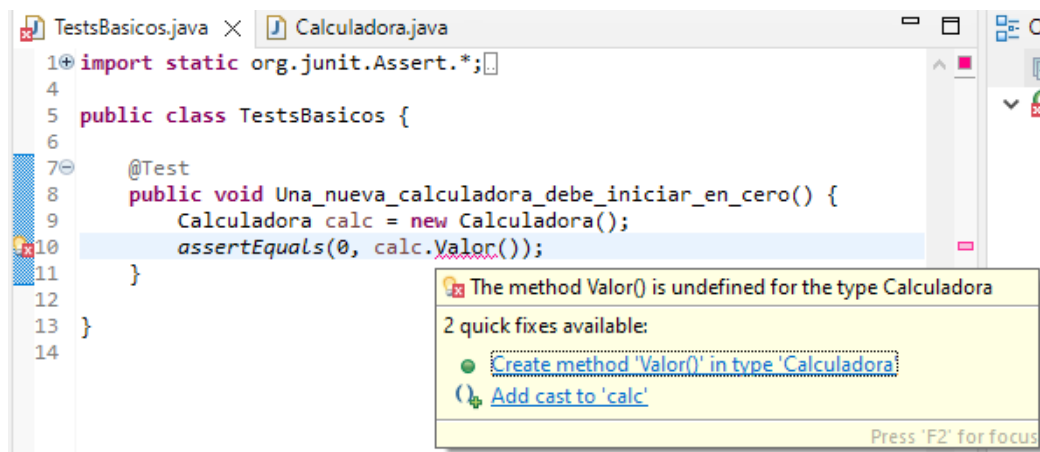
2. Habilitamos el control de versiones con Git.



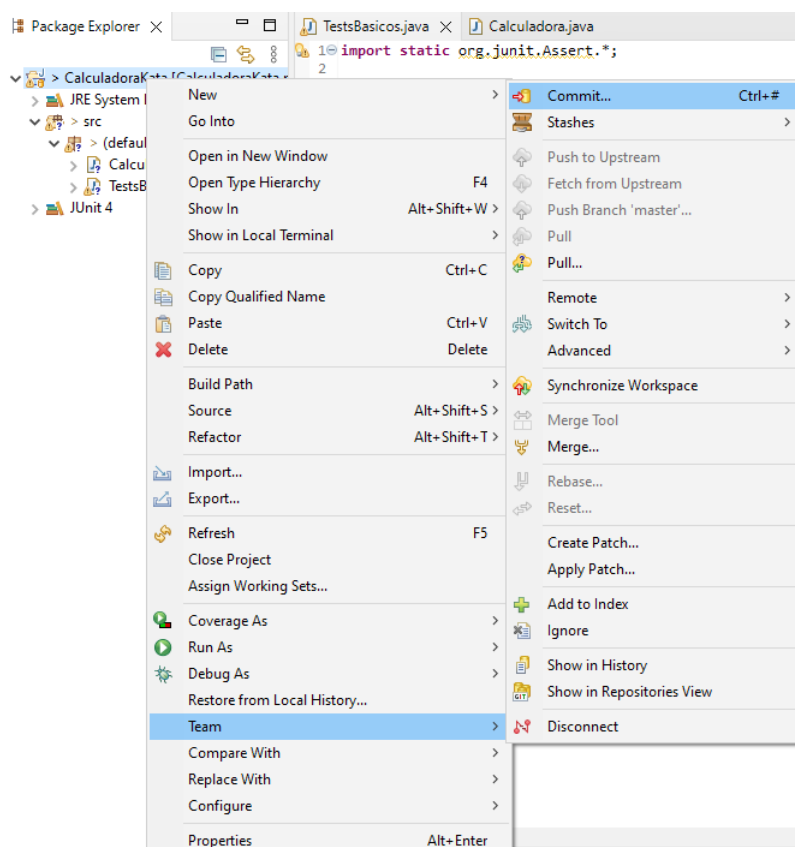
3. Creamos primer método. Posicionamos el cursor sobre "Calculadora" para crear la clase necesaria.

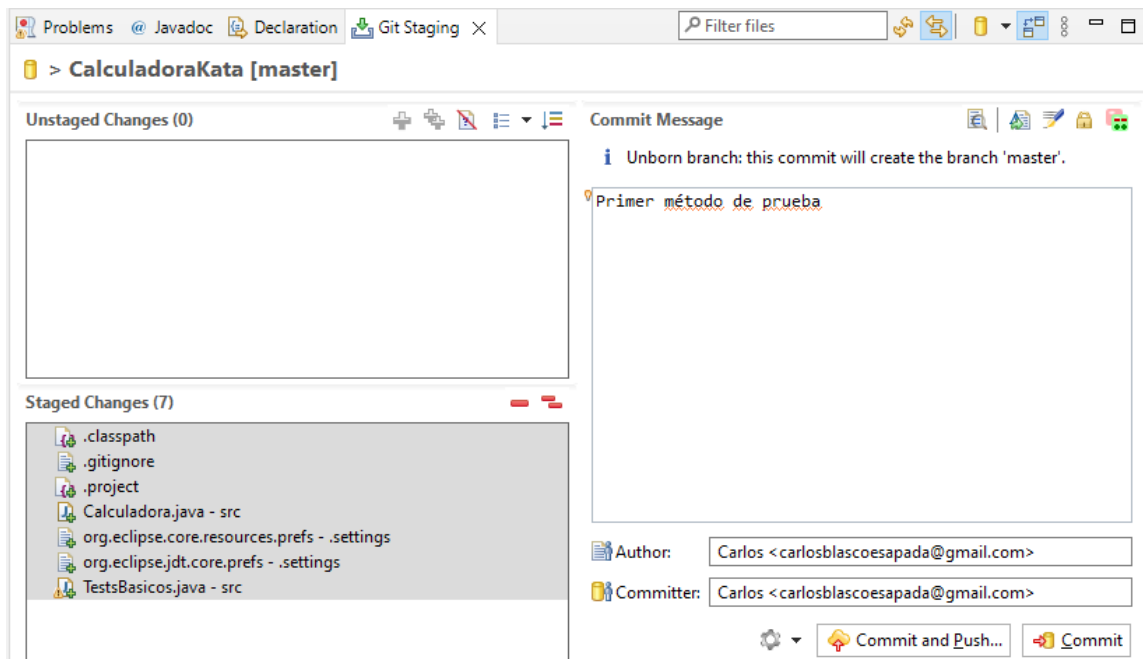


4. Repetimos el proceso con "Valor". Debemos cambiar Object por "int" y null por "1".

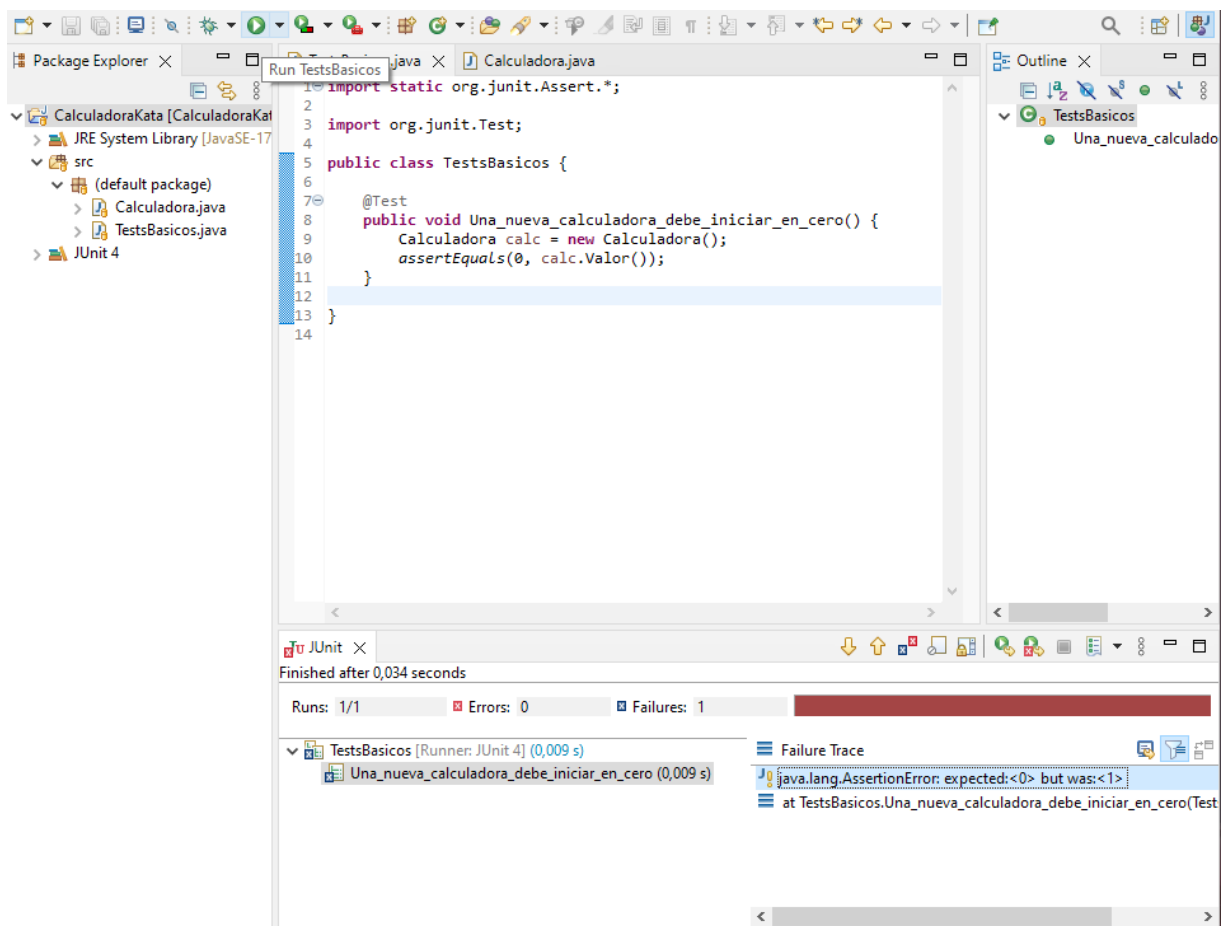


5. Realizamos un commit con los cambios. Añadimos los archivos con el doble "+" verde.

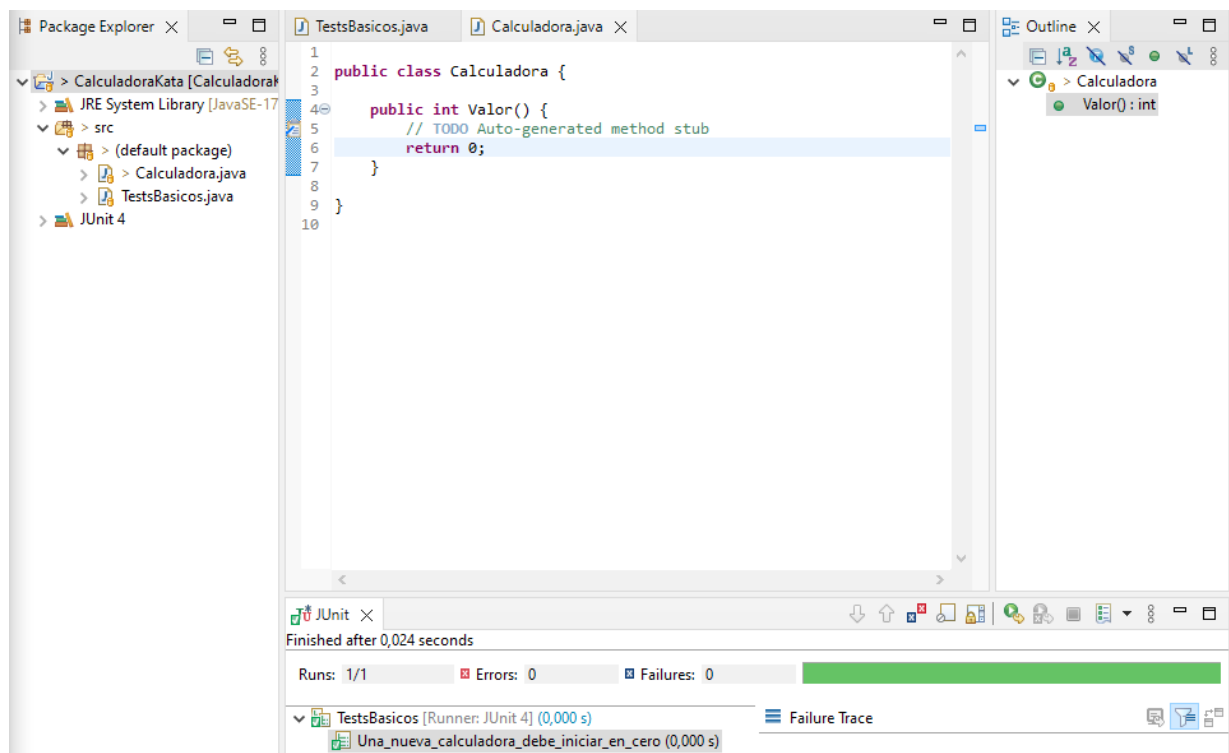




6. Ejecutamos el primer test con el botón de ejecutar al igual que en IntelliJ IDEA. El test será erróneo ya que esperaba un valor de 0 y recibe 1 (return 1;).



7. Al cambiar el valor devuelto por 0 el test será exitoso.



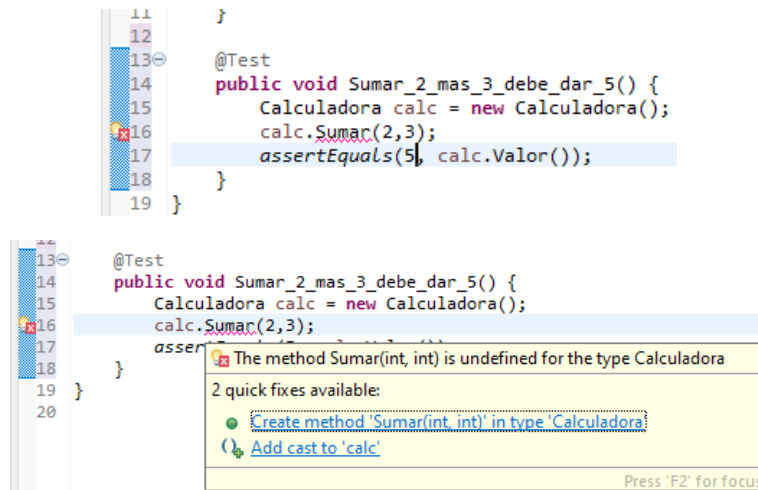
8. Realizamos un commit del estado actual. Seleccionamos todos los archivos.

The screenshot displays an IDE interface with the following components:

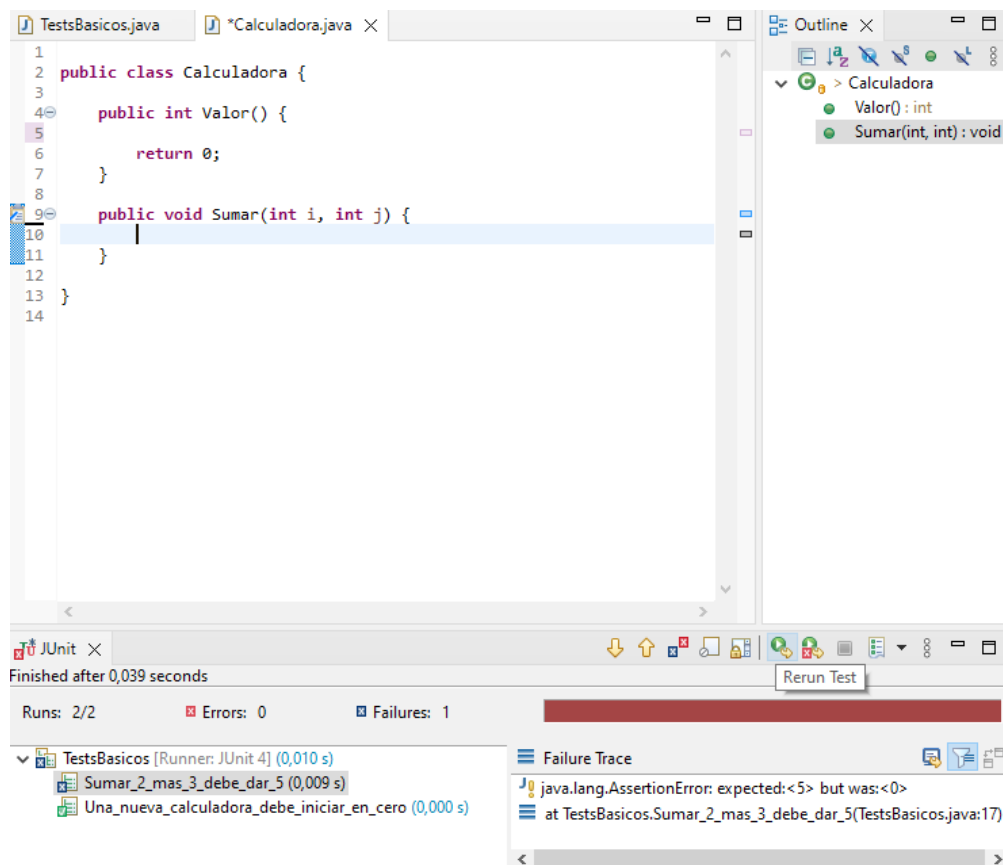
- Editor:** Shows `TestsBasicos.java` with the following code:

```
1 import static org.junit.Assert.*;
2
3 import org.junit.Test;
4
5 public class TestsBasicos {
6
7     @Test
8     public void Una_nueva_calculadora_debe_iniciar_en_cero() {
9         Calculadora calc = new Calculadora();
10        assertEquals(0, calc.Valor());
11    }
12
13 }
14
```
- Outline:** Shows the project structure with `TestsBasicos` and `Una_nueva_calculado`.
- JUnit Runner:** Shows the test results for `TestsBasicos` (Runner: JUnit 4) with a duration of 0,000 s. The test `Una_nueva_calculadora_debe_iniciar_en_cero` (0,000 s) passed successfully.
- Git Staging:** Shows the commit interface for the `CalculadoraKata [master]` branch. The `Unstaged Changes` section is empty. The `Staged Changes` section contains two files: `Calculadora.java - src` and `TestsBasicos.java - src`. The `Commit Message` field contains the text `Corregido primer método`. The `Author` and `Committer` fields are both set to `Carlos <carlosblascoesapada@gmail.com>`. The `Commit and Push...` button is highlighted.

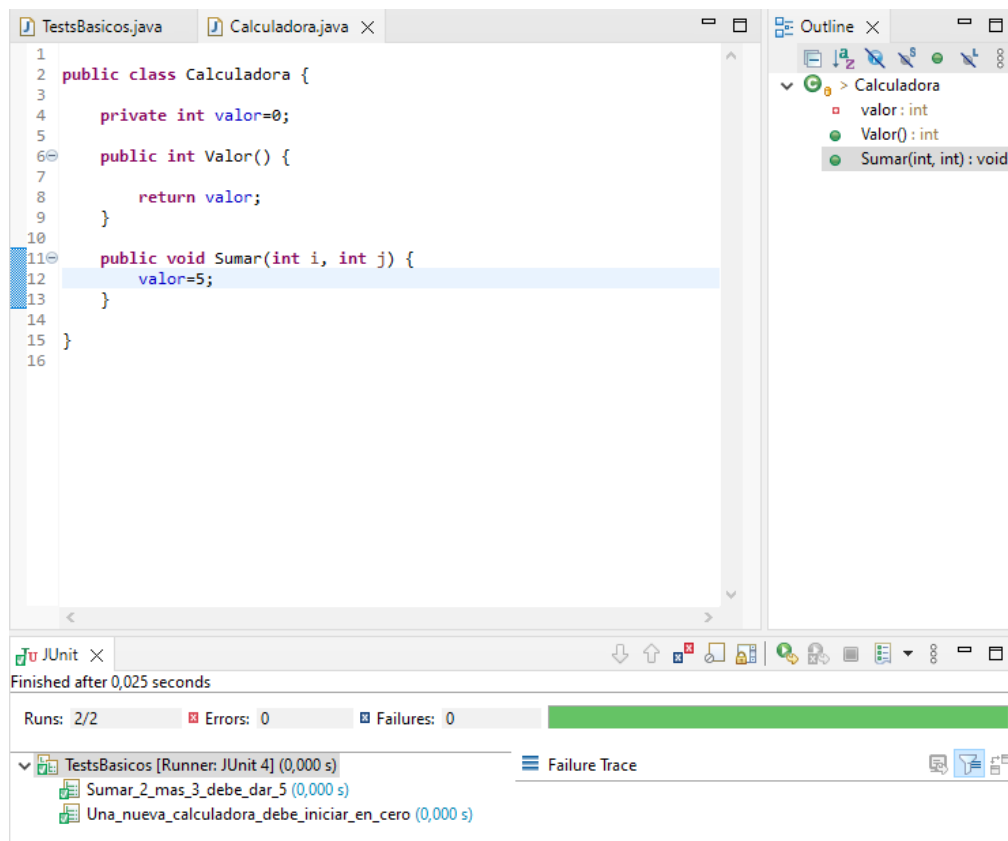
9. Creamos un nuevo método para probar la suma ($2 + 3$ espera 5 como resultado). Seleccionamos "Sumar" para crear el atributo en la clase Coche automáticamente.



10. Ejecutamos el test con el botón indicado en la captura (Rerun Test) y lo pasamos incorrectamente, esperaba 5 pero fue 0.



11. Aplicamos la lógica para modificar la clase Calculadora añadiendo el atributo "valor". Al realizar el test con este código sale correcto.



The screenshot shows an IDE with two tabs: TestsBasicos.java and Calculadora.java. The Calculadora.java file contains the following code:

```
1 public class Calculadora {
2
3     private int valor=0;
4
5     public int Valor() {
6
7         return valor;
8     }
9
10    public void Sumar(int i, int j) {
11        valor=5;
12    }
13
14 }
15
16
```

The Outline view on the right shows the class structure:

- Calculadora
 - valor: int
 - Valor(): int
 - Sumar(int, int): void

The bottom panel shows the test results for TestsBasicos [Runner: JUnit 4] (0,000 s). The tests passed:

- Sumar_2_mas_3_debe_dar_5 (0,000 s)
- Una_nueva_calculadora_debe_iniciar_en_cero (0,000 s)

The status bar indicates: Finished after 0,025 seconds, Runs: 2/2, Errors: 0, Failures: 0.

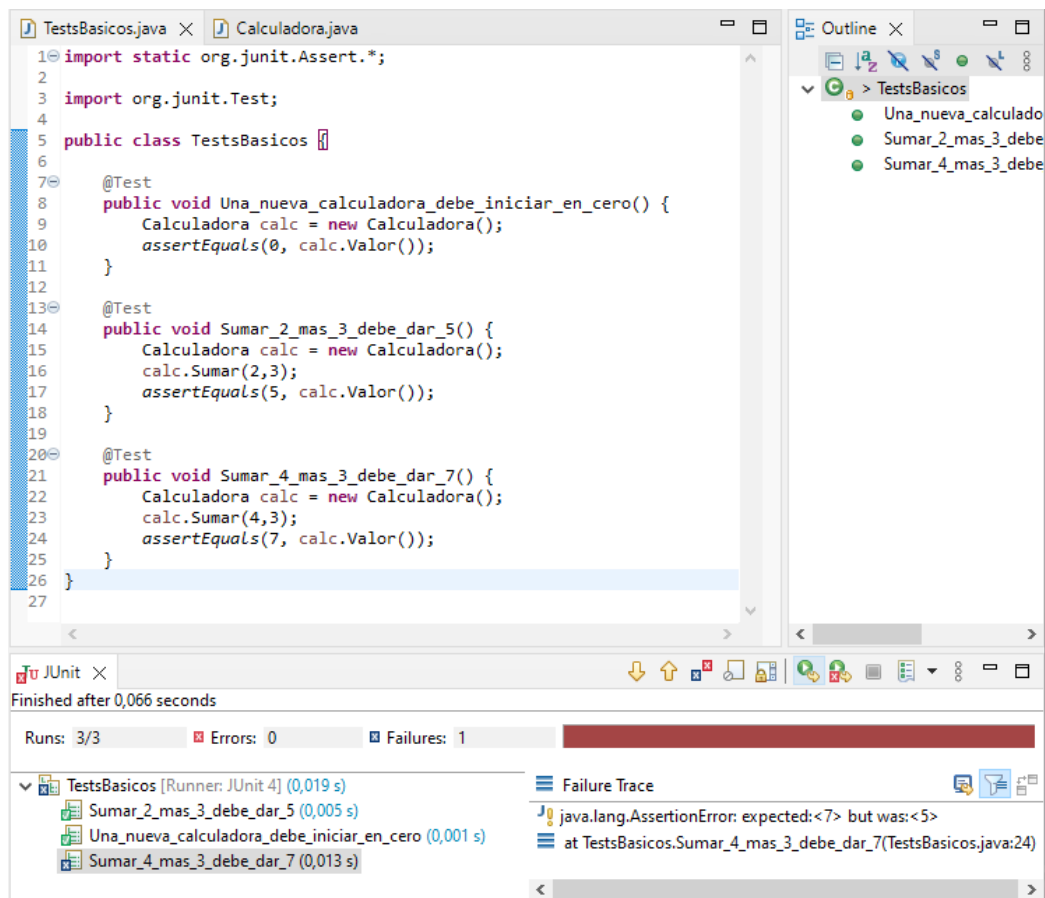
12. Realizamos un commit con los cambios.

The screenshot displays an IDE interface with the following components:

- Editor:** Shows `TestsBasicos.java` with two test methods:

```
1 import static org.junit.Assert.*;
2
3 import org.junit.Test;
4
5 public class TestsBasicos {
6
7     @Test
8     public void Una_nueva_calculadora_debe_iniciar_en_cero() {
9         Calculadora calc = new Calculadora();
10        assertEquals(0, calc.Valor());
11    }
12
13    @Test
14    public void Sumar_2_mas_3_debe_dar_5() {
15        Calculadora calc = new Calculadora();
16        calc.Sumar(2,3);
17        assertEquals(5, calc.Valor());
18    }
19 }
20
```
- Outline:** Lists `TestsBasicos` with sub-items `Una_nueva_calculadora_debe_iniciar_en_cero` and `Sumar_2_mas_3_debe_dar_5`.
- JUnit Runner:** Shows "Finished after 0,025 seconds", "Runs: 2/2", "Errors: 0", and "Failures: 0".
- Git Staging:** Shows "CalculadoraKata [master]" with "Unstaged Changes (0)" and "Staged Changes (2)" including `Calculadora.java - src` and `TestsBasicos.java - src`.
- Commit Message:** Contains the text "Nuevo método de prueba".
- Author/Committer:** Both fields are filled with "Carlos <carlosblascoesapada@gmail.com>".
- Buttons:** "Commit and Push..." and "Commit".

13. Creamos un nuevo método de prueba con la suma de 4 y 3, al realizar el test da error ya que espera 7 pero recibe 5.



```
1 import static org.junit.Assert.*;
2
3 import org.junit.Test;
4
5 public class TestsBasicos {
6
7     @Test
8     public void Una_nueva_calculadora_debe_iniciar_en_cero() {
9         Calculadora calc = new Calculadora();
10        assertEquals(0, calc.Valor());
11    }
12
13    @Test
14    public void Sumar_2_mas_3_debe_dar_5() {
15        Calculadora calc = new Calculadora();
16        calc.Sumar(2,3);
17        assertEquals(5, calc.Valor());
18    }
19
20    @Test
21    public void Sumar_4_mas_3_debe_dar_7() {
22        Calculadora calc = new Calculadora();
23        calc.Sumar(4,3);
24        assertEquals(7, calc.Valor());
25    }
26 }
27
```

JUnit ×
Finished after 0,066 seconds

Runs: 3/3 Errors: 0 Failures: 1

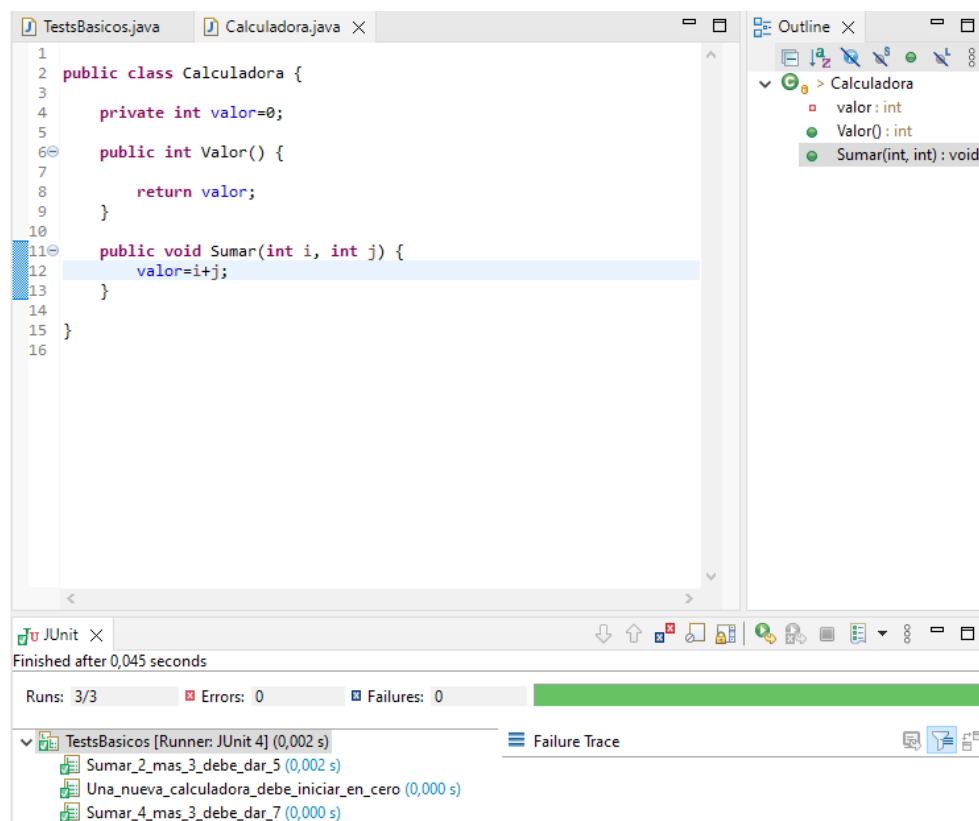
TestsBasicos [Runner: JUnit 4] (0,019 s)

- Sumar_2_mas_3_debe_dar_5 (0,005 s)
- Una_nueva_calculadora_debe_iniciar_en_cero (0,001 s)
- Sumar_4_mas_3_debe_dar_7 (0,013 s)

Failure Trace

java.lang.AssertionError: expected:<7> but was:<5>
at TestsBasicos.Sumar_4_mas_3_debe_dar_7(TestsBasicos.java:24)

14. Modificamos el método Sumar para que no devuelva un valor fijo, sino que sume ambas variables. Rerun Test nuevamente y salen todos correctos.



```
1 public class Calculadora {
2
3     private int valor=0;
4
5     public int Valor() {
6
7         return valor;
8     }
9
10
11    public void Sumar(int i, int j) {
12        valor=i+j;
13    }
14
15 }
16
```

JUnit ×
Finished after 0,045 seconds

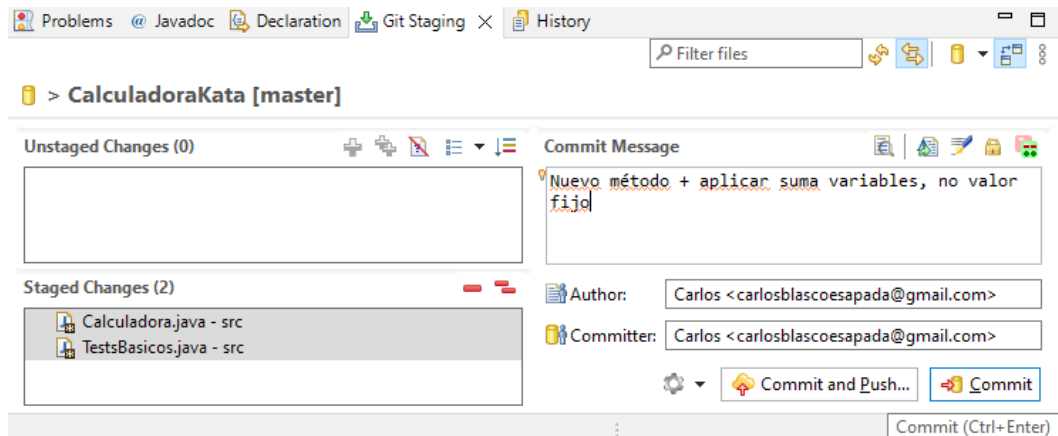
Runs: 3/3 Errors: 0 Failures: 0

TestsBasicos [Runner: JUnit 4] (0,002 s)

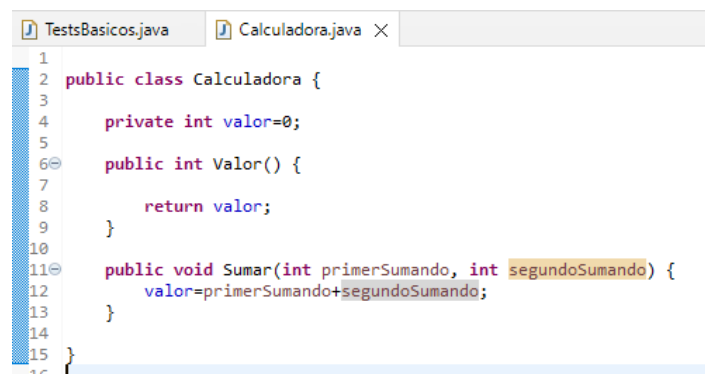
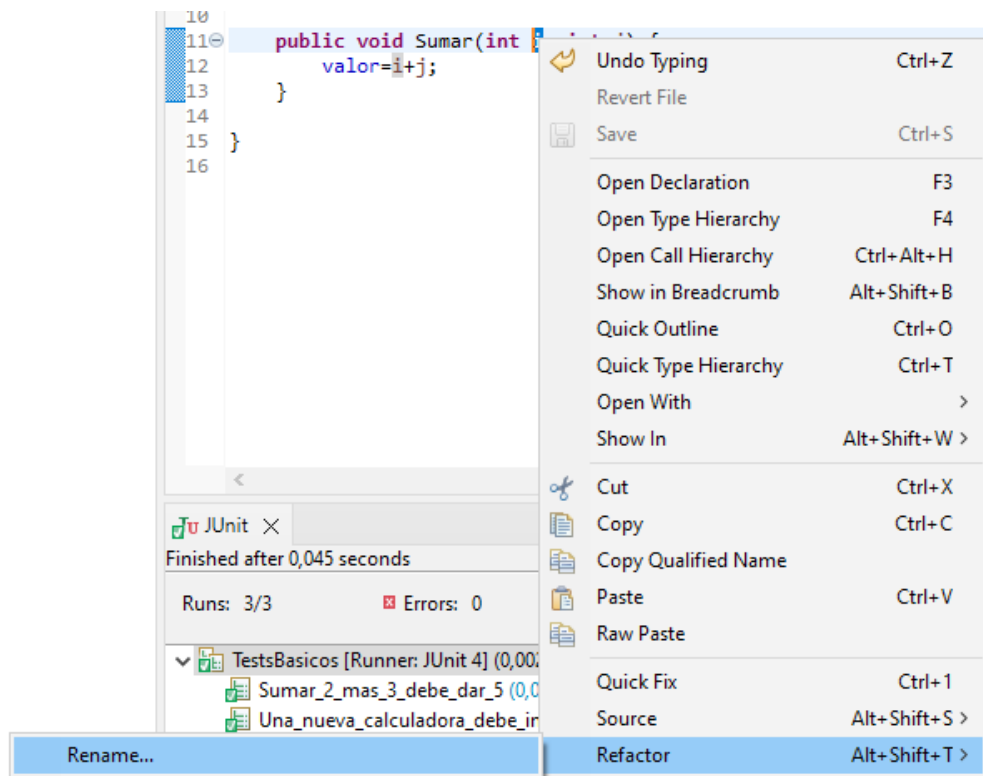
- Sumar_2_mas_3_debe_dar_5 (0,002 s)
- Una_nueva_calculadora_debe_iniciar_en_cero (0,000 s)
- Sumar_4_mas_3_debe_dar_7 (0,000 s)

Failure Trace

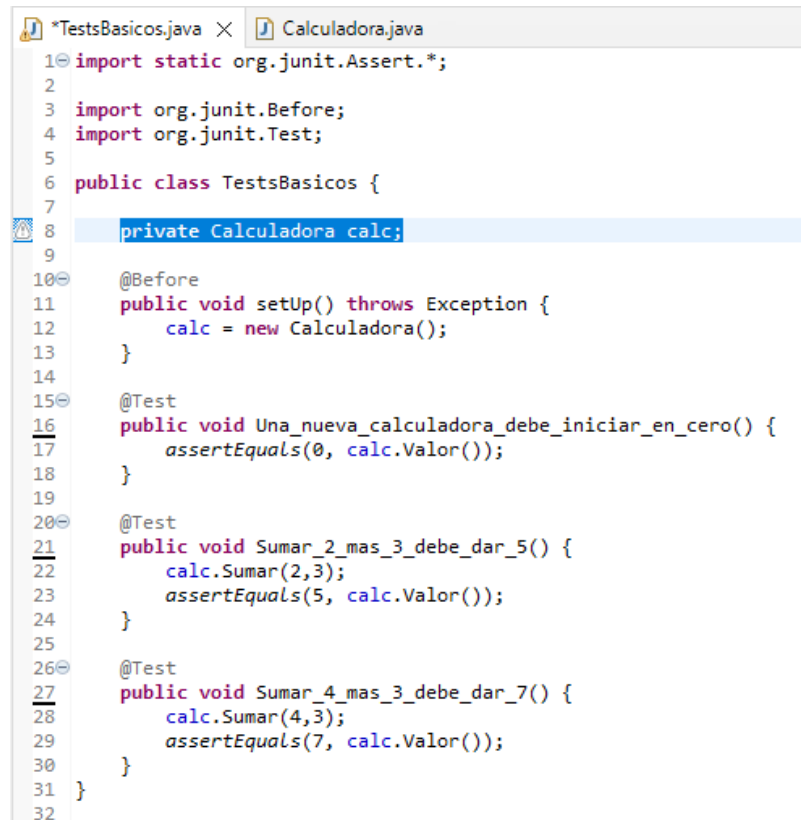
15. Hacemos un commit con los cambios.



16. Refactorizamos las variables para Sumar de esta forma:

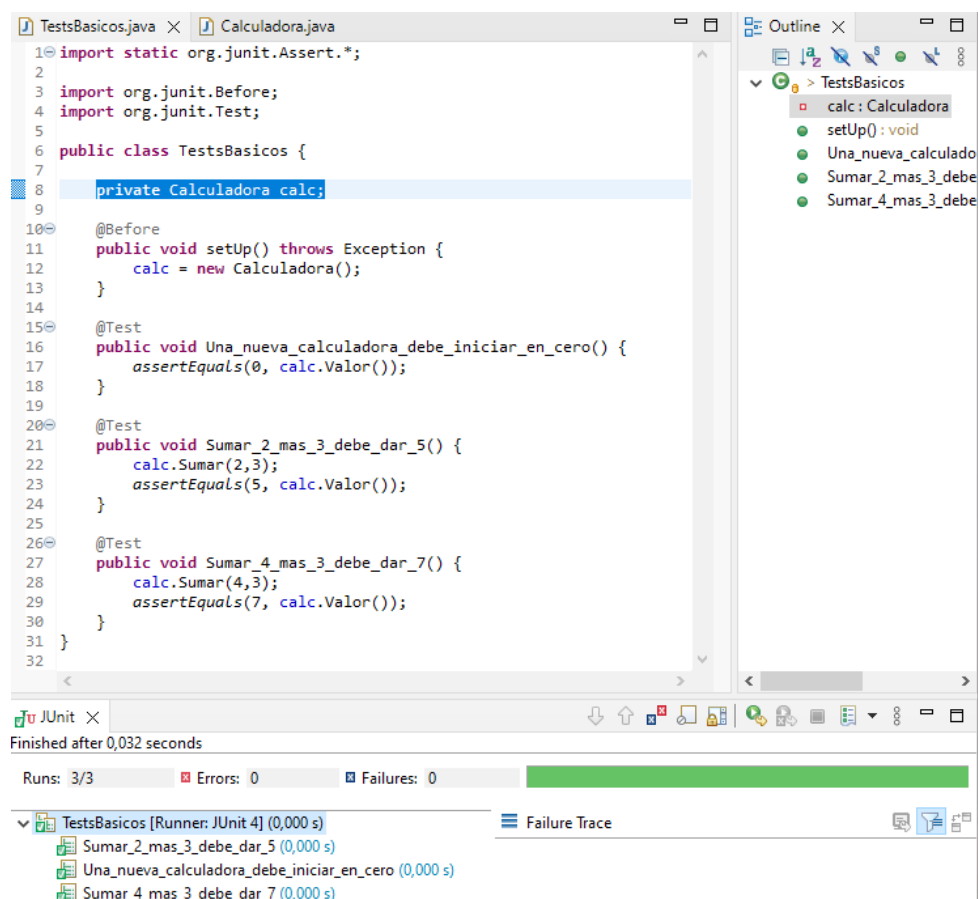


17. Además, modificamos el test para que se instancie una única calculadora, en vez de una calculadora por método (solo habrá un "Calculadora calc = new Calculadora();").



```
1 import static org.junit.Assert.*;
2
3 import org.junit.Before;
4 import org.junit.Test;
5
6 public class TestsBasicos {
7
8     private Calculadora calc;
9
10    @Before
11    public void setUp() throws Exception {
12        calc = new Calculadora();
13    }
14
15    @Test
16    public void Una_nueva_calculadora_debe_iniciar_en_cero() {
17        assertEquals(0, calc.Valor());
18    }
19
20    @Test
21    public void Sumar_2_mas_3_debe_dar_5() {
22        calc.Sumar(2,3);
23        assertEquals(5, calc.Valor());
24    }
25
26    @Test
27    public void Sumar_4_mas_3_debe_dar_7() {
28        calc.Sumar(4,3);
29        assertEquals(7, calc.Valor());
30    }
31 }
32
```

18. Pasamos el test con éxito.



```
1 import static org.junit.Assert.*;
2
3 import org.junit.Before;
4 import org.junit.Test;
5
6 public class TestsBasicos {
7
8     private Calculadora calc;
9
10    @Before
11    public void setUp() throws Exception {
12        calc = new Calculadora();
13    }
14
15    @Test
16    public void Una_nueva_calculadora_debe_iniciar_en_cero() {
17        assertEquals(0, calc.Valor());
18    }
19
20    @Test
21    public void Sumar_2_mas_3_debe_dar_5() {
22        calc.Sumar(2,3);
23        assertEquals(5, calc.Valor());
24    }
25
26    @Test
27    public void Sumar_4_mas_3_debe_dar_7() {
28        calc.Sumar(4,3);
29        assertEquals(7, calc.Valor());
30    }
31 }
32
```

JUnit 4

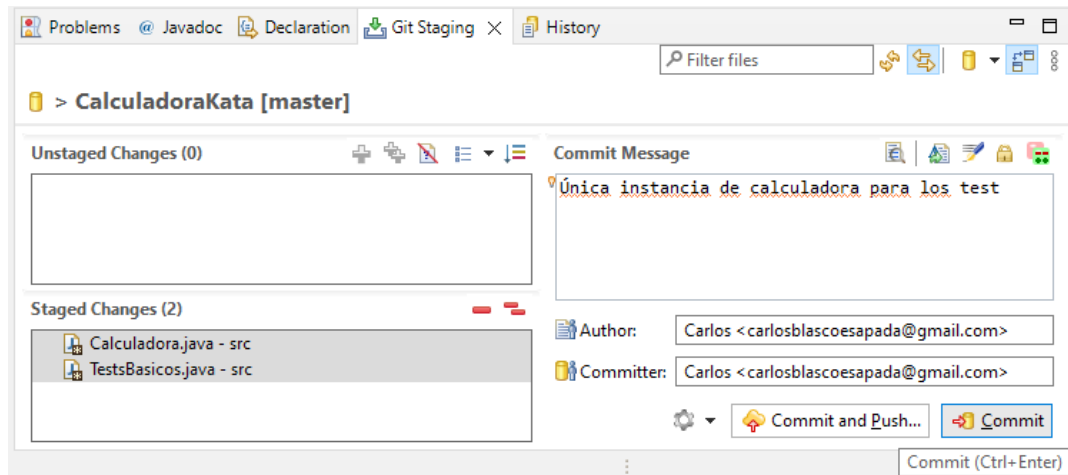
Finished after 0,032 seconds

Runs: 3/3 Errors: 0 Failures: 0

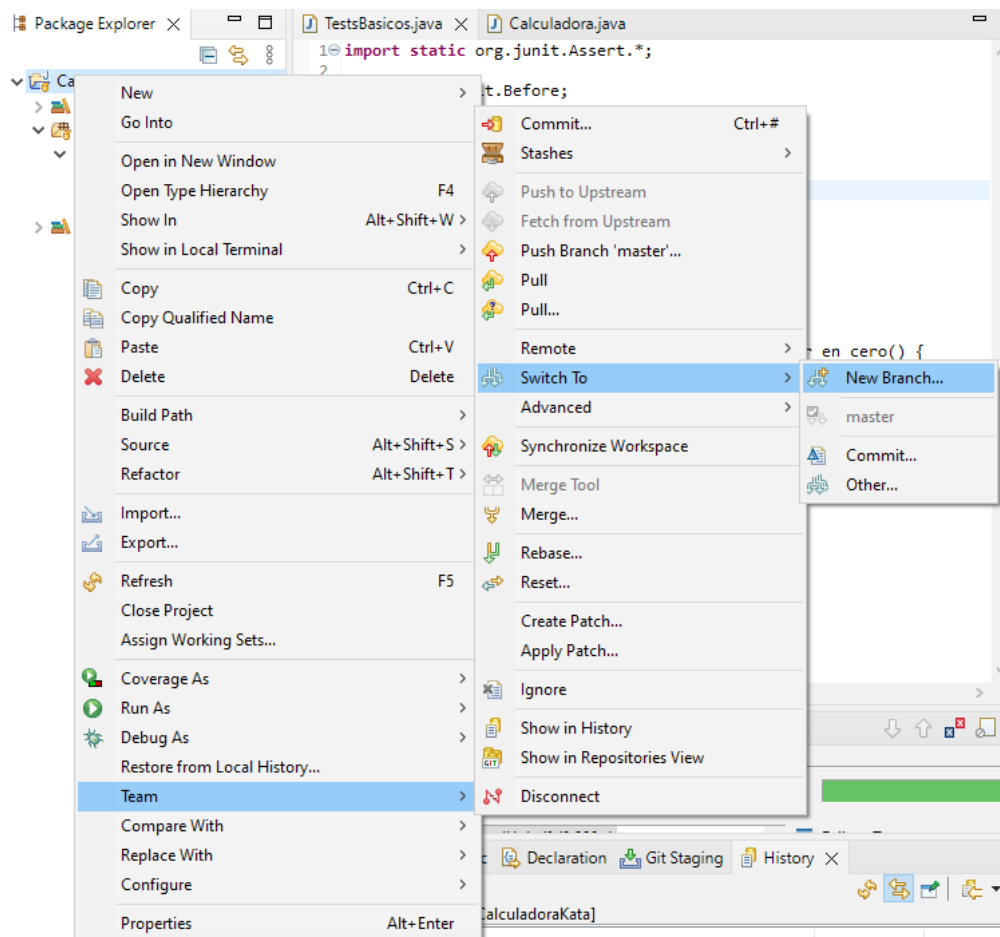
TestsBasicos [Runner: JUnit 4] (0,000 s)

- Sumar_2_mas_3_debe_dar_5 (0,000 s)
- Una_nueva_calculadora_debe_iniciar_en_cero (0,000 s)
- Sumar_4_mas_3_debe_dar_7 (0,000 s)

19. Hacemos el commit.




20. Creamos la rama Refactorizado y nos situamos en ella. Clic derecho en el proyecto y seguir los pasos de la captura.



Create Branch

Create a new branch in repository CalculadoraKata

Please choose a source branch and a name for the new branch

Source:  master

Select...

Branch name:

☐ Configure upstream for push and pull

When pulling:

☒ Check out new branch

Finish

Cancel

Package Explorer

CalculadoraKata [CalculadoraKata]

JRE System Library [JavaSE-17]

src

(default package)

Calculadora.java

TestsBasicos.java

JUnit 4

TestsBasicos.java

Calculadora.java

```
1 import static org.junit.Assert.*;
2
3 import org.junit.Before;
4 import org.junit.Test;
5
6 public class TestsBasicos {
7
8     private Calculadora calc;
9
10    @Before
11    public void setUp() throws Exception {
12        calc = new Calculadora();
13    }
14
15    @Test
16    public void Una_nueva_calculadora_debe_iniciar_en_cero() {
17        assertEquals(0, calc.Valor());
18    }
19
20    @Test
21    public void Sumar_2_mas_3_debe_dar_5() {
22        calc.Sumar(2,3);
23    }
24 }
```

Problems

Javadoc

Declaration

Git Staging

History

JUnit

Filter files

CalculadoraKata [Refactorizado]

Unstaged Changes (0)

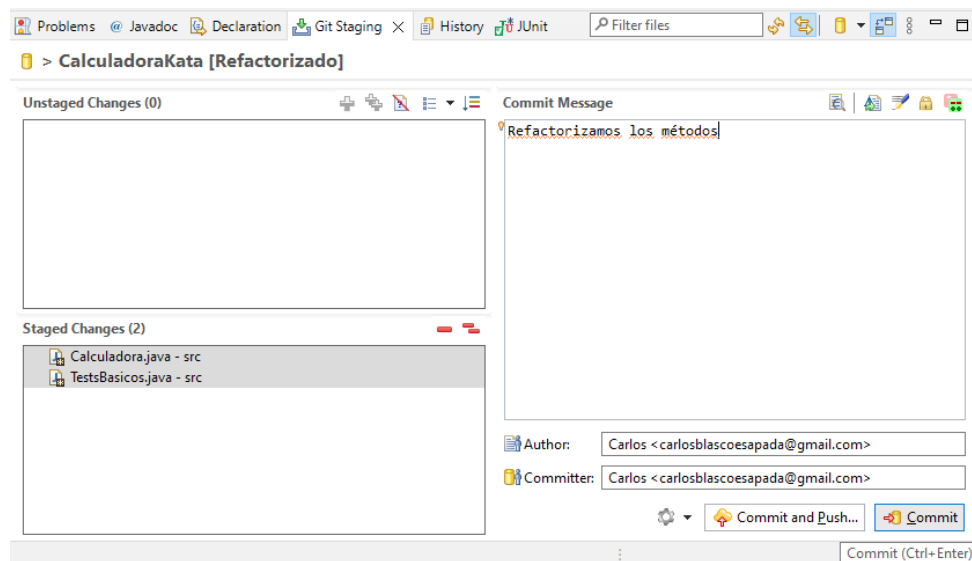
Commit Message

21. Refactorizamos de la siguiente manera (al igual que en el paso 16):

```
TestsBasicos.java x Calculadora.java
3 import org.junit.Before;
4 import org.junit.Test;
5
6 public class TestsBasicos {
7
8     private Calculadora calc;
9
10    @Before
11    public void setUp() throws Exception {
12        calc = new Calculadora();
13    }
14
15    @Test
16    public void Una_nueva_calculadora_debe_iniciar_en_cero_CarlosBlascoEspada()
17        assertEquals(0, calc.Valor());
18    }
19
20    @Test
21    public void Sumar_2_mas_3_debe_dar_5_CarlosBlascoEspada() {
22        calc.Sumar(2,3);
23        assertEquals(5, calc.Valor());
24    }
25
26    @Test
27    public void Sumar_4_mas_3_debe_dar_7_CarlosBlascoEspada() {
28        calc.Sumar(4,3);
29        assertEquals(7, calc.Valor());
30    }
```

```
TestsBasicos.java Calculadora.java x
1
2 public class Calculadora {
3
4     private int valor=0;
5
6     public int Valor() {
7
8         return valor;
9     }
10
11    public void Sumar_CarlosBlascoEspada(int primerSumando, int segundoSumando)
12        valor=primerSumando+segundoSumando;
13    }
14
15 }
16
```

22. Realizamos un commit con los cambios.



23. Siguiendo los mismos pasos creamos una nueva rama llamada Memoria, en la cual subiremos este pdf.