



El futuro digital
es de todos

MinTIC

JUnit

Documentación y Configuración
JUNIT - Parte 2

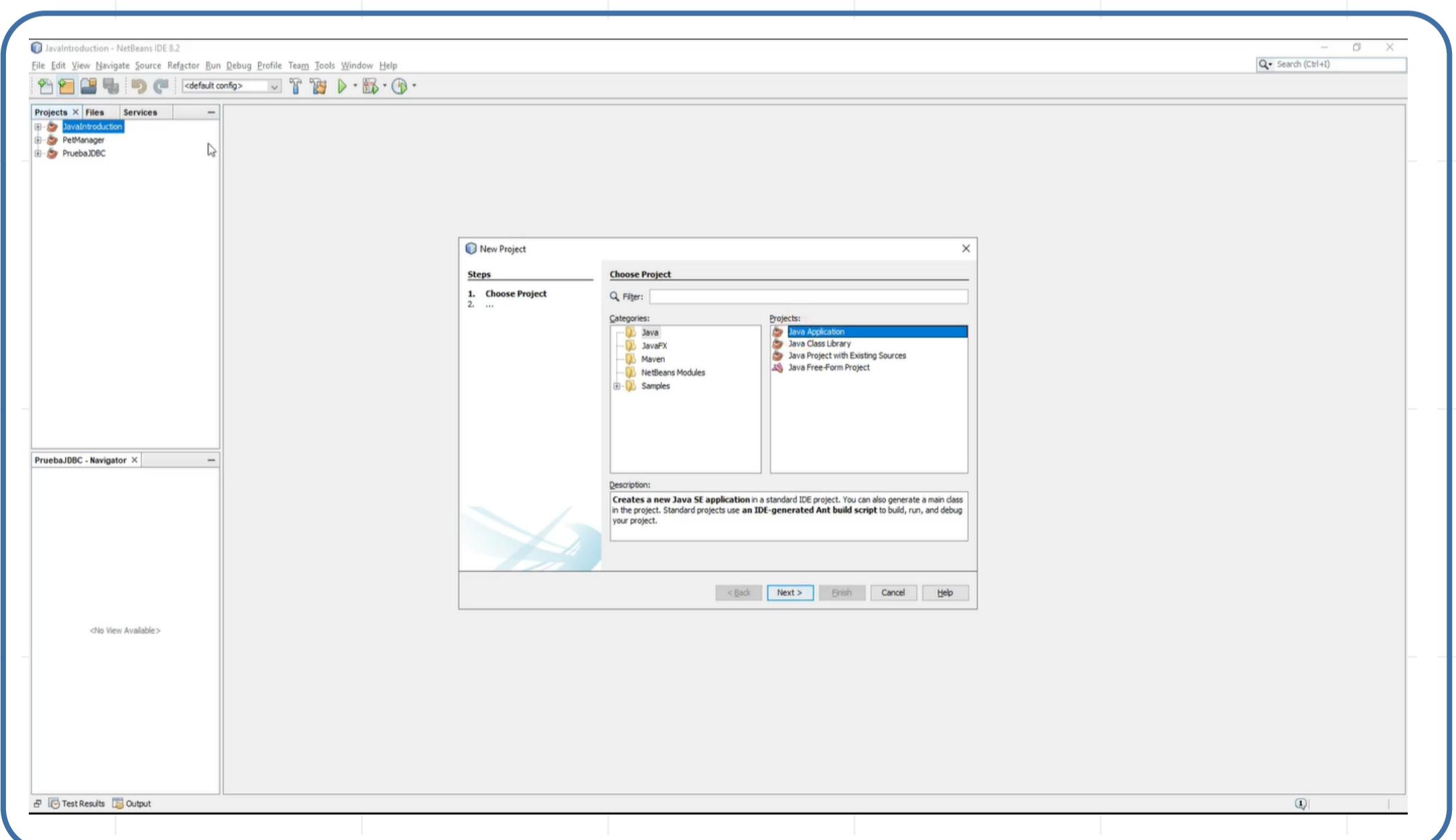


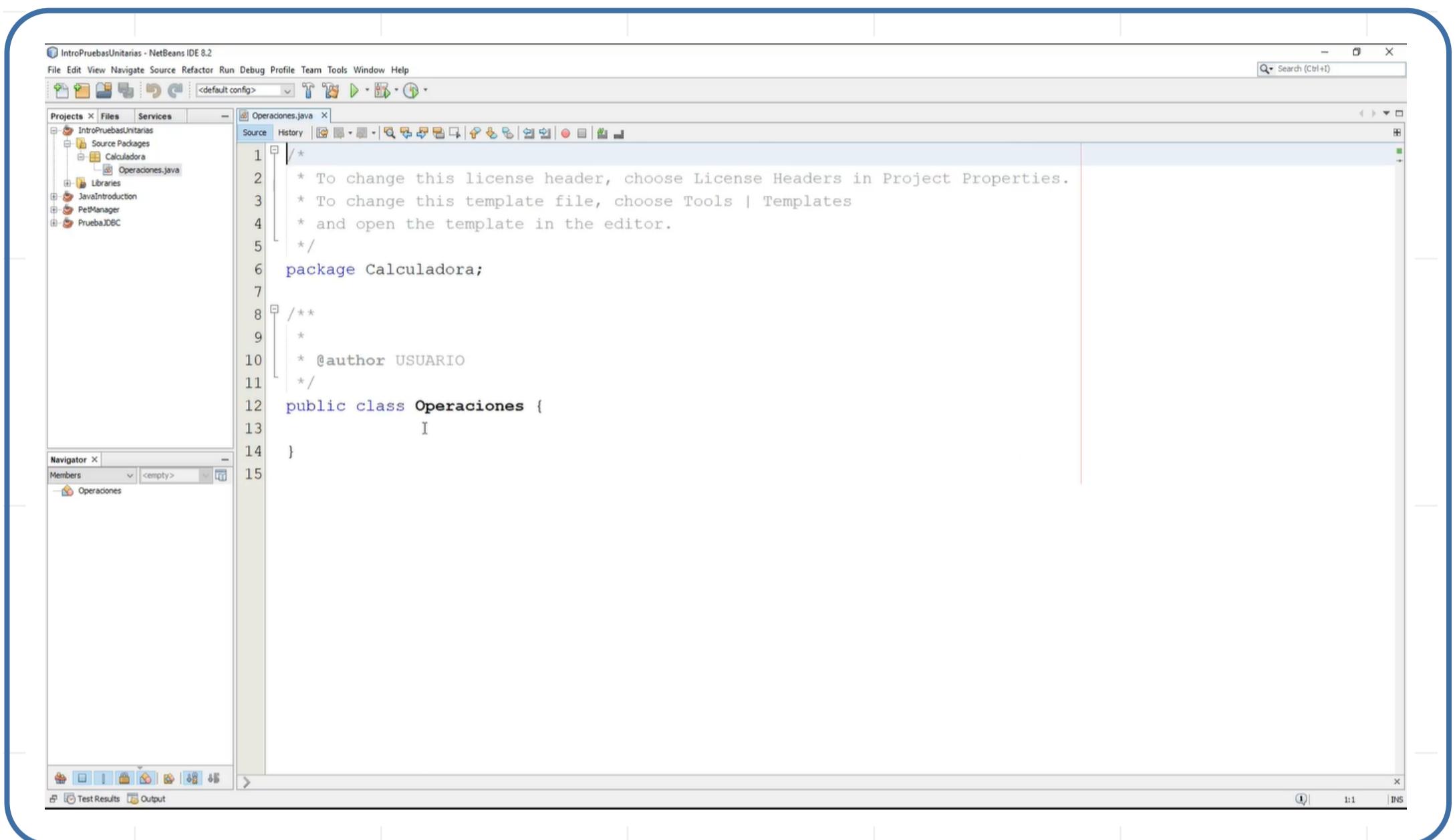
Universidad de Caldas

Hola:

Para continuar con el concepto de **calidad de software** se realizará un ejemplo de implementación de pruebas unitarias con JUNIT, donde se dispuso de la documentación y todo el concepto de esta librería. A continuación se mostrará el desarrollo de unas clases con métodos de operaciones matemáticas que ayudarán en el proceso de aprendizaje. También cómo integrar JUNIT al proyecto, cómo agregar archivos de prueba y conocer los componentes básicos de dichos archivos.

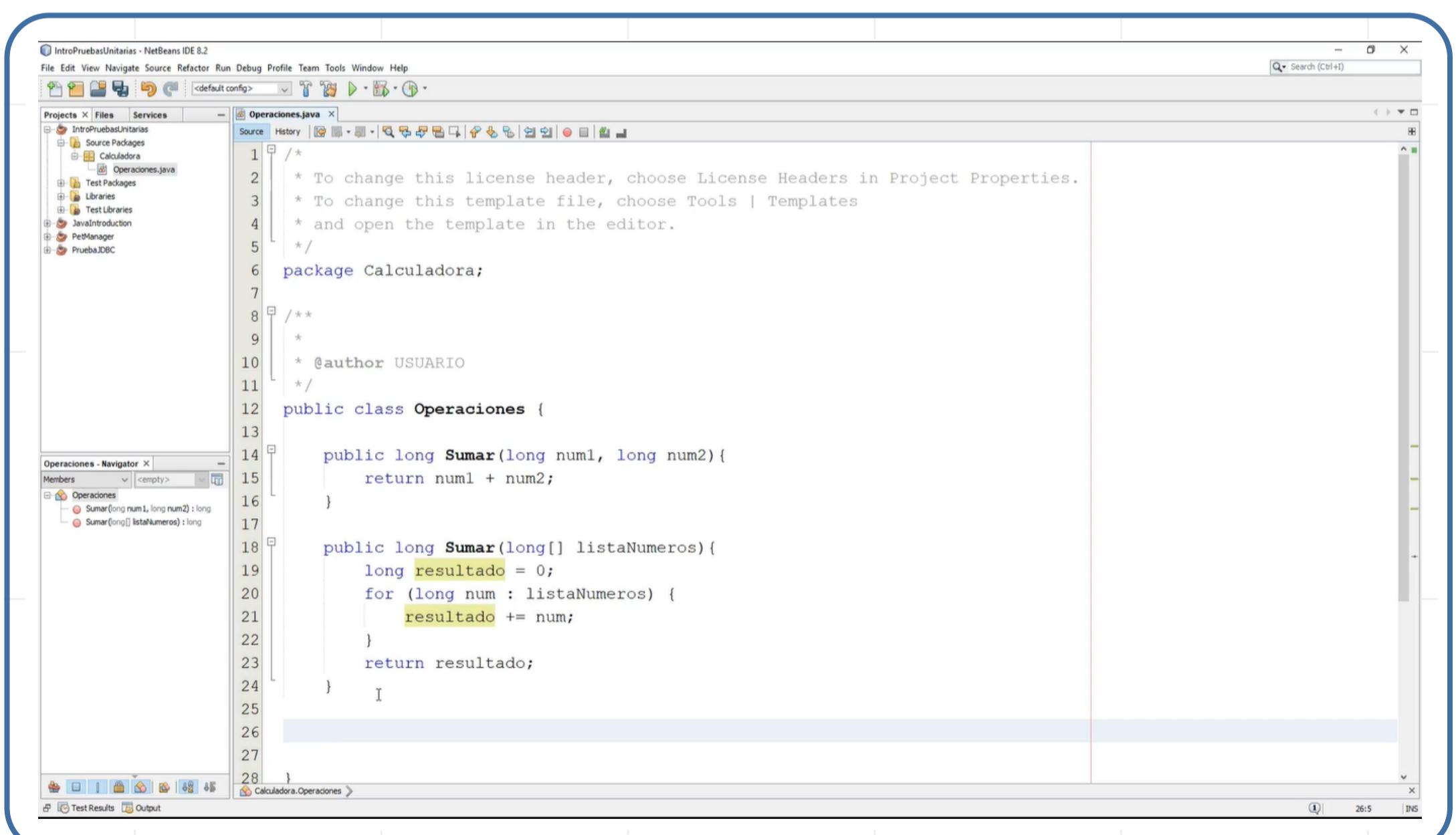
Video de explicación de JUNIT y todos sus componentes de manera práctica.





The screenshot shows the NetBeans IDE interface with the title bar "IntroPruebasUnitarias - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has icons for New Project, Open Project, Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others. The Projects tab is selected in the left sidebar, showing a project named "IntroPruebasUnitarias" with a "Source Packages" node containing a "Calculadora" node and an "Operaciones.java" file. The Files tab is also visible. The main editor window displays the "Operaciones.java" source code:

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6 package Calculadora;
7
8 /**
9 *
10 * @author USUARIO
11 */
12 public class Operaciones {
13
14 }
15
```



The screenshot shows the NetBeans IDE interface with the title bar "IntroPruebasUnitarias - NetBeans IDE 8.2". The menu bar and toolbar are identical to the previous screenshot. The Projects tab is selected in the left sidebar. The main editor window displays the "Operaciones.java" source code, which now includes two methods:

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6 package Calculadora;
7
8 /**
9 *
10 * @author USUARIO
11 */
12 public class Operaciones {
13
14     public long Sumar(long num1, long num2){
15         return num1 + num2;
16     }
17
18     public long Sumar(long[] listaNumeros){
19         long resultado = 0;
20         for (long num : listaNumeros) {
21             resultado += num;
22         }
23         return resultado;
24     }
25
26
27
28 }
```

A Navigator panel titled "Operaciones - Navigator" is open on the left, showing the members of the "Operaciones" class: "Sumar(long num1, long num2) : long" and "Sumar(long[] listaNumeros) : long". The status bar at the bottom right shows "26:5" and "INS".

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** IntroPruebasUnitarias - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar with icons for file operations.
- Project Explorer:** Shows the project structure with a single source package "Operaciones" containing a class "Operaciones.java".
- Source Editor:** Displays the Java code for the "Operaciones" class. The code includes methods for addition, subtraction, multiplication, division, and handling lists of numbers. A try-catch block for division is highlighted.
- Navigator:** Shows the members of the "Operaciones" class, including methods like Dividir, Multiplicar, Restar, and Sumar.
- Status Bar:** Shows the current file path (Calculadora.Operaciones), line number (44), and column number (32).

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** IntroPruebasUnitarias - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar with icons for file operations.
- Project Explorer:** Shows the project structure with a single source package "Operaciones" containing a class "Operaciones.java".
- Source Editor:** Displays the Java code for the "Operaciones" class, specifically focusing on the "Multiplicar" method.
- New File Dialog:** A modal dialog titled "New File" is open, showing the "Choose File Type" step. It lists categories like Java, JavaFX, and Swing GUI Forms, and file types like JUnit Test, Test for Existing Class, and Test Suite. The "Test for Existing Class" option is selected.
- Status Bar:** Shows the current file path (Calculadora.Operaciones), line number (31), and column number (28).

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** IntroPruebasUnitarias - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar with icons for file operations.
- Project Explorer:** Shows the project structure under "IntroPruebasUnitarias".
 - Source Packages: Calculadora (with Operaciones.java), Test Packages: <default package> (with PruebasOperaciones.java), Libraries: JUnit 4.12 - junit-4.12.jar, Hamcrest 1.3 - hamcrest-core-1.3.jar.
 - JavaIntroduction, PetManager, PruebaJDBC are also listed.
- Code Editor:** The "PruebasOperaciones.java" file is open, displaying Java test code using JUnit annotations.

```
18 public class PruebasOperaciones {  
19  
20     public PruebasOperaciones() {  
21         }  
22  
23     @BeforeClass  
24     public static void setUpClass() {  
25         }  
26  
27     @AfterClass  
28     public static void tearDownClass() {  
29         }  
30  
31     @Before  
32     public void setUp() {  
33         }  
34  
35     @After  
36     public void tearDown() {  
37         }  
38  
39         // TODO add test methods here.  
40         // The methods must be annotated with annotation @Test. For example:  
41         //  
42         // @Test  
43         // public void hello() {}  
44     }  
45 }
```
- Navigator:** Shows the members of the PruebasOperaciones class, including the constructor and the four annotated methods: setUp(), setUpClass(), tearDown(), and tearDownClass().
- Status Bar:** Test Results, Output, and other status indicators.

The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** IntroPruebasUnitarias - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar with icons for file operations.
- Project Explorer:** Shows the project structure under "IntroPruebasUnitarias".
 - Source Packages: Calculadora (containing Operaciones.java)
 - Test Packages: <default package> (containing PruebasOperaciones.java)
 - Libraries: JUnit 4.12 - junit-4.12.jar, Hamcrest 1.3 - hamcrest-core-1.3.jar
- Code Editor:** The "PruebasOperaciones.java" file is open, displaying Java code for unit tests.

```
43 //  
44 // @Test  
45 // public void hello() {}  
46  
47 @Test  
48 public void Prueba1() {  
49     I  
50 }  
51  
52 @Test  
53 public void Prueba2() {  
54 }  
55  
56 @Test  
57 public void Prueba3() {  
58 }  
59  
60 }  
61  
62 }
```

A yellow circle highlights the opening brace of the `Prueba2()` method at line 54.
- Navigator:** The "Prueba3 - Navigator" window shows the members of the `PruebasOperaciones` class:
 - Members: PruebasOperaciones(), Prueba1(), Prueba2(), Prueba3(), setUp(), setUpClass(), tearDown(), tearDownClass()
- Bottom Status Bar:** Test Results, Output, and other status indicators.

The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer:** Shows the project "IntroPruebasUnitarias" with packages like "Source Packages" (containing "Operaciones.java") and "Test Packages" (containing "PruebasOperaciones.java"). Libraries "JUnit 4.12 - junit-4.12.jar" and "Hamcrest 1.3 - hamcrest-core-1.3.jar" are listed.
- Code Editor:** Displays the file "PruebasOperaciones.java" with Java code for JUnit tests. It includes annotations @BeforeClass, @AfterClass, and @Before, and prints messages to System.out.
- Output Window:** Shows the "Test Results" for "OperacionesTest". It indicates "Tests passed: 100,00 %" and "All 3 tests passed. (0,065 s)".
- Console Output:** Prints the following text:

```
Método único antes de la clase
Inicio de método
Fin de método
Inicio de método
Fin de método
Inicio de método
Fin de método
Método al finalizar la clase
```

Con esto se ha aprendido cómo agregar JUNIT a un proyecto Java con Netbeans y cómo crear archivos de prueba.



**Mision
TIC2022**



Universidad de Caldas