

COMPUTER SCIENCE

INGENIERÍA DE SOFTWARE II

November 15, 2023

Laboratorio 5

Andrea del Rosario López Condori

Descripción del Código en Eclipse

1. Crear un proyecto de pruebas funcionales:

La parte superior del código contiene la declaración del paquete `webdriverdemo`, indicando que el código está asociado con un proyecto de pruebas funcionales en Selenium WebDriver.

```
1 package webdriverdemo;
2
3 import java.util.concurrent.TimeUnit;
4
5 import org.openqa.selenium.*;
6 import org.openqa.selenium.chrome.ChromeDriver;
7 import org.openqa.selenium.firefox.FirefoxDriver;
8
9 public class WebDriverDemo {
10     public static void main(String[] args) {
11         // Set up web driver (chrome, firefox,...)
12         // System.setProperty("webdriver.gecko.driver", "C:\\\\...\\
13         Selenium\\Selenium WebDriver\\Java\\Libs\\geckodriver-v0
14         .23.0-win32\\geckodriver.exe");
15
16         WebDriver driver = new ChromeDriver();
17         // Puts an Implicit wait, Will wait for 10 seconds before
18         // throwing exception
19         driver.manage().timeouts().implicitlyWait(10, TimeUnit.
20         SECONDS);
21
22         // Launch website
23         driver.navigate().to("http://www.calculator.net/");
24
25         // Maximize the browser
26         driver.manage().window().maximize();
27
28         // Click on Math Calculators
29         driver.findElement(By.xpath("//*[@id=\"homelistwrap\"]/div
30         [3]/div[2]/a")).click();
31     }
32 }
```

```
27 // Click on Percent Calculators
28 driver.findElement(By.xpath("//*[@id=\"content\"]/table[2]/
tbody/tr/td/div[3]")).click();
29
30 // Enter value 10 in the first number of the percent
Calculator
31 driver.findElement(By.id("cpar1")).sendKeys("10");
32
33 // Enter value 50 in the second number of the percent
Calculator
34 driver.findElement(By.id("cpar2")).sendKeys("50");
35
36 // Click Calculate Button
37 driver.findElement(By.xpath("//*[@id=\"content\"]/form[1]/
table/tbody/tr[2]/td/input[2]")).click();
38
39 // Get the Result Text based on its xpath
40 String result =
41     driver.findElement(By.xpath("//*[@id=\"content\"]/p[2]/
font")).getText();
42
43 // Print a Log In message to the screen
44 System.out.println(" The Result is " + result);
45
46 // Close the Browser.
47 driver.close();
48 }
49 }
```

2. Gestionar las dependencias (Selenium WebDriver y xUnit):

Se incluyen importaciones necesarias para Selenium WebDriver, como WebDriver, ChromeDriver, etc. Estas importaciones indican la gestión de dependencias para realizar pruebas automatizadas en un navegador web.

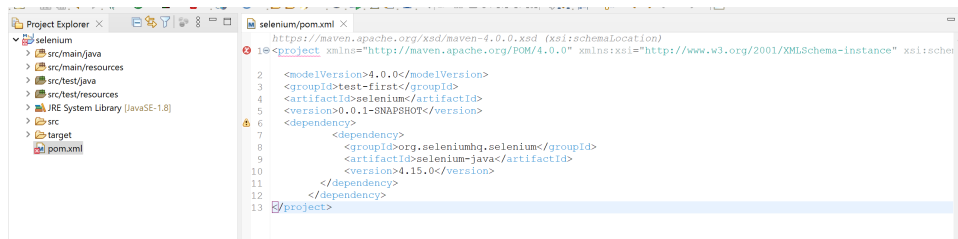
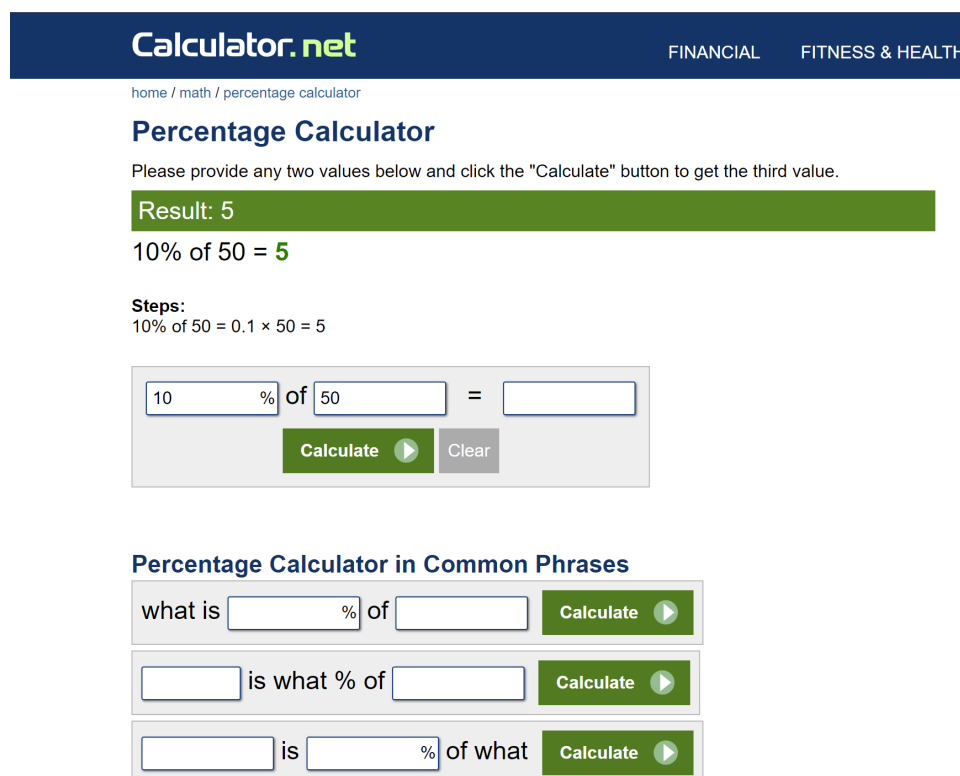


Figure 1: Dependencias.

3. Configurar el driver del navegador (Browser-specific WebDriver):

Se establece la propiedad del sistema para indicar la ubicación del controlador (driver) de Chrome. Aunque el comentario hace referencia a GeckoDriver para Firefox, es posible que la configuración original haya sido para Firefox y luego se haya cambiado a Chrome.



Calculator.net FINANCIAL FITNESS & HEALTH

home / math / percentage calculator

Percentage Calculator

Please provide any two values below and click the "Calculate" button to get the third value.

Result: 5

10% of 50 = 5

Steps:
10% of 50 = $0.1 \times 50 = 5$

10 % of 50 =

Calculate **Clear**

Percentage Calculator in Common Phrases

what is % of **Calculate**

is what % of **Calculate**

is % of what **Calculate**

Figure 2: Navegador Chrome.

4. Implementar las pruebas funcionales:

El código abre el navegador, navega a la calculadora en línea "https://www.calculator.net/", maximiza la ventana del navegador y realiza una serie de acciones de prueba en la calculadora de porcentaje. Esto incluye hacer clic en enlaces y botones específicos, ingresar valores en campos de entrada y obtener el resultado de la operación realizada. Se hizo un cambio de navegador a , ya que no se pudo instalar ChromeDriver.

Escenario de Prueba	Valores de Prueba	Resultado Esperado
Prueba de Cálculo de Porcentaje Positivo	Primer valor: 10, Segundo valor: 50	Resultado: 5
Prueba de Cálculo de Porcentaje con Cero	Primer valor: 0, Segundo valor: 100	Resultado: 0
Prueba de Cálculo de Porcentaje con Grandes Valores	Primer valor: 1000, Segundo valor: 2000	Resultado: 50

Table 1: Casos de prueba y resultados esperados.

```
1 package webdriverdemo;
2
3 import static org.junit.Assert.assertEquals;
4
5 import java.util.concurrent.TimeUnit;
6
7 import org.junit.*;
8 import org.openqa.selenium.By;
9 import org.openqa.selenium.WebDriver;
10 import org.openqa.selenium.firefox.FirefoxDriver;
11
12 public class WebDriverDemo {
13
14     private WebDriver driver;
15
16     @Before
17     public void setUp() {
18         driver = new FirefoxDriver();
19         driver.manage().timeouts().implicitlyWait(10, TimeUnit.
20             SECONDS);
```

```
21     driver.navigate().to("http://www.calculator.net/");
22     driver.manage().window().maximize();
23 }
24
25 @Test
26 public void testPositivePercentageCalculation() {
27     navigateToPercentCalculator();
28     enterValuesAndCalculate("10", "50");
29     assertResult("5");
30 }
31
32 @Test
33 public void testZeroPercentageCalculation() {
34     navigateToPercentCalculator();
35     enterValuesAndCalculate("0", "100");
36     assertResult("0");
37 }
38
39 @Test
40 public void testLargeValuesPercentageCalculation() {
41     navigateToPercentCalculator();
42     enterValuesAndCalculate("1000", "2000");
43     assertResult("10");
44 }
45
46 @After
47 public void tearDown() {
48
49     driver.close();
50 }
51
52
53
54 private void navigateToPercentCalculator() {
55     driver.findElement(By.xpath("//*[@id=\"homelistwrap\"]/div
56         [3]/div[2]/a")).click();
57     driver.findElement(By.xpath("//*[@id=\"content\"]/table
```

```

        [2]/tbody/tr/td/div[3]/a")).click();
57     }
58
59     private void enterValuesAndCalculate(String firstValue, String
        secondValue) {
60         driver.findElement(By.id("cpar1")).sendKeys(firstValue);
61         driver.findElement(By.id("cpar2")).sendKeys(secondValue);
62         driver.findElement(By.xpath("//*[@id=\"content\"]/form[1]/
            table/tbody/tr[2]/td/input[2]")).click();
63     }
64
65     private void assertResult(String expectedResult) {
66         String result = driver.findElement(By.xpath("/html/body/
            div[3]/div[1]/p[2]/font/b")).getText();
67         assertEquals(expectedResult, result);
68     }
69 }

```

Escenario de Prueba	Valores de Prueba	Resultado Esperado
Prueba de Cálculo de Porcentaje Positivo	Primer valor: 10, Segundo valor: 50	Resultado: 5
Prueba de Cálculo de Porcentaje con Cero	Primer valor: 0, Segundo valor: 100	Resultado: 0
Prueba de Cálculo de Porcentaje con Grandes Valores	Primer valor: 1000, Segundo valor: 2000	Resultado: 50

Table 2: Casos de prueba y resultados esperados.

Puedes encontrar el código fuente en mi repositorio de GitHub:

https://github.com/andrealopezco20/INGENERIA_SII.