

¿Cómo implementar POM?

POM sencillo:

Es la estructura básica del marco del modelo de objetos de página donde todos los elementos web del **AUT** y los métodos que operan en estos elementos web se mantienen dentro de un archivo de clase. Una tarea como la verificación debe estar separada como parte de los métodos de prueba.

```
public class Guru99Login {  
    WebDriver driver;  
    By user99GuruName = By.name("uid");  
    By password99Guru = By.name("password");  
    By titleText = By.className("barone");  
    By login = By.name("btnLogin");  
  
    public Guru99Login(WebDriver driver){  
        this.driver = driver;  
    }  
    //Set user name in textbox  
    public void setUserName(String strUserName){  
        driver.findElement(user99GuruName).sendKeys(strUserName);  
    }  
}
```

1 Page class in object repository

Find Web Element

Performing operation on web element

2

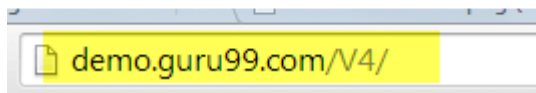
3

Ejemplo completo

Caso de prueba: Vaya al sitio de demostración de Guru99.

Paso 1) Vaya al sitio de demostración de Guru99

Goto Guru99 Demo site

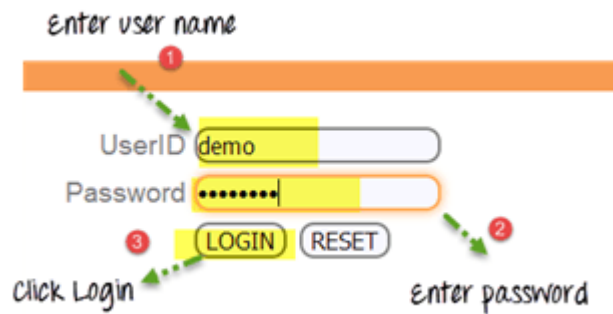


Paso 2) En la página de inicio, verifique que el texto "Guru99 Bank" esté presente

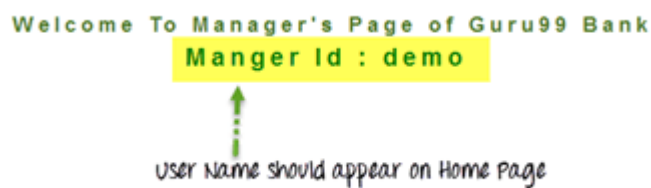
Verify if you found this title
'Guru99 Bank' in Login Page

Guru99 Bank

Paso 3) Inicie sesión en la aplicación



Paso 4) Verifique que la página de inicio contenga texto como "Id. del administrador: demostración"



Aquí estamos tratando con 2 páginas.

1. Página de inicio
2. Página de inicio (se muestra una vez que inicia sesión)

En consecuencia, creamos 2 POM en Selenium privadas

Página de inicio de sesión de Guru99 POM

```
package pages;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
public class Guru99Login {
```

```
    WebDriver driver;
```

```
    By user99GuruName = By.name("uid");
```

```
    By password99Guru = By.name("password");
```

```
By titleText =By.className("barone");
```

```
By login = By.name("btnLogin");
```

```
public Guru99Login(WebDriver driver){
```

```
    this.driver = driver;
```

```
}
```

```
//Set user name in textbox
```

```
public void setUsername(String strUserName){
```

```
    driver.findElement(user99GuruName).sendKeys(strUserName);
```

```
}
```

```
//Set password in password textbox
```

```
public void setPassword(String strPassword){
```

```
    driver.findElement(password99Guru).sendKeys(strPassword);
```

```
}
```

```
//Click on login button
```

```
public void clickLogin(){
```

```
        driver.findElement(login).click();

    }

    //Get the title of Login Page

    public String getLoginTitle(){

        return  driver.findElement(titleText).getText();

    }

    /**

    * This POM method will be exposed in test case to login in the application

    * @param strUserName

    * @param strPasword

    * @return

    */

    public void loginToGuru99(String strUserName,String strPasword){

        //Fill user name

        this.setUserName(strUserName);

        //Fill password
```

```
        this.setPassword(strPasword);

        //Click Login button

        this.clickLogin();
    }

}
```

Página de inicio de Guru99 POM en Selenium

```
package pages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

public class Guru99HomePage {

    WebDriver driver;

    By homePageUserName = By.xpath("//table//tr[@class='heading3']");

    public Guru99HomePage(WebDriver driver){

        this.driver = driver;

    }

    //Get the User name from Home Page
```

```
public String getHomePageDashboardUserName(){  
  
    return driver.findElement(homePageUserName).getText();  
  
}  
  
}
```

Guru99 POM simple en Selenium Caso de prueba

```
package test;  
  
import java.util.concurrent.TimeUnit;  
  
import org.openqa.selenium.WebDriver;  
  
import org.openqa.selenium.firefox.FirefoxDriver;  
  
import org.testng.Assert;  
  
import org.testng.annotations.BeforeTest;  
  
import org.testng.annotations.Test;  
  
import pages.Guru99HomePage;  
  
import pages.Guru99Login;  
  
public class Test99GuruLogin {  
  
    String driverPath = "C:\\geckodriver.exe";
```

```
WebDriver driver;
```

```
Guru99Login objLogin;
```

```
Guru99HomePage objHomePage;
```

```
@BeforeTest
```

```
public void setup(){
```

```
    System.setProperty("webdriver.gecko.driver", driverPath);
```

```
    driver = new FirefoxDriver();
```

```
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

```
    driver.get("https://demo.guru99.com/V4/");
```

```
}
```

```
/**
```

```
 * This test case will login in https://demo.guru99.com/V4/
```

```
 * Verify login page title as guru99 bank
```

```
 * Login to application
```

```
 * Verify the home page using Dashboard message
```

```
 */
```

```

@Test(priority=0)

public void test_Home_Page_Appear_Correct(){

    //Create Login Page object

    objLogin = new Guru99Login(driver);

    //Verify login page title

    String loginPageTitle = objLogin.getLoginTitle();

    Assert.assertTrue(loginPageTitle.toLowerCase().contains("guru99 bank"));

    //login to application

    objLogin.loginToGuru99("mgr123", "mgr!23");

    // go the next page

    objHomePage = new Guru99HomePage(driver);

    //Verify home page

    Assert.assertTrue(objHomePage.getHomePageDashboardUserName().toLowerCase().contains("manger id : mgr123"));

}

```

AD

¿Qué es Page Factory en Selenium?

Fábrica de páginas en Selenium es un concepto de marco de modelo de objetos de página incorporado para Selenium WebDriver pero está muy optimizado. Se utiliza para la inicialización de objetos de página o para crear una instancia del propio objeto de página. También se utiliza para inicializar elementos de clase de página sin utilizar "FindElement/s".

Aquí también seguimos el concepto de separación entre el repositorio de objetos de página y los métodos de prueba. Además, con la ayuda de la clase PageFactory en Selenium, usamos anotaciones **@BuscarPor** para encontrar WebElement. Usamos el método `initElements` para inicializar elementos web.

```
@FindBy(xpath="//table//tr[@class='heading3']")
WebElement homePageUserName;

public Guru99HomePage(WebDriver driver){
    this.driver = driver;
    //This initElements method will create all WebElements
    PageFactory.initElements(driver, this);
}
```

@BuscarPor poder aceptar **nombre de etiqueta, texto de enlace parcial, nombre, texto de enlace, id, css, nombre de clase, xpath** como atributos.

Veamos el mismo ejemplo anterior usando Page Factory

Página de inicio de sesión de Guru99 con Page Factory

```
package PageFactory;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.support.FindBy;
```

```
import org.openqa.selenium.support.PageFactory;
```

```
public class Guru99Login {
```

```
/**
```

* All WebElements are identified by @FindBy annotation

*/

WebDriver driver;

@FindBy(name="uid")

WebElement user99GuruName;

@FindBy(name="password")

WebElement password99Guru;

@FindBy(className="barone")

WebElement titleText;

@FindBy(name="btnLogin")

WebElement login;

public Guru99Login(WebDriver driver){

 this.driver = driver;

 //This initElements method will create all WebElements

 PageFactory.initElements(driver, this);

```
}
```

```
//Set user name in textbox
```

```
public void setUsername(String strUserName){
```

```
    user99GuruName.sendKeys(strUserName);
```

```
}
```

```
//Set password in password textbox
```

```
public void setPassword(String strPassword){
```

```
    password99Guru.sendKeys(strPassword);
```

```
}
```

```
//Click on login button
```

```
public void clickLogin(){
```

```
    login.click();
```

```
}
```

```
//Get the title of Login Page
```

```
public String getLoginTitle(){
```

```
    return titleText.getText();
```

```
}

/**

 * This POM method will be exposed in test case to login in the application

 * @param strUserName

 * @param strPassword

 * @return

 */

public void loginToGuru99(String strUserName,String strPassword){

    //Fill user name

    this.setUserName(strUserName);

    //Fill password

    this.setPassword(strPassword);

    //Click Login button

    this.clickLogin();

}

}
```

Página de inicio de Guru99 con Page Factory

```
package PageFactory;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;

public class Guru99HomePage {

    WebDriver driver;

    @FindBy(xpath="//table//tr[@class='heading3']")

    WebElement homePageUserName;

    public Guru99HomePage(WebDriver driver){

        this.driver = driver;

        //This initElements method will create all WebElements

        PageFactory.initElements(driver, this);

    }

    //Get the User name from Home Page
```

```

    public String getHomePageDashboardUserName(){

        return  homePageUserName.getText();

    }

}

```

Guru99 TestCase con el concepto Page Factory

```

package test;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.Assert;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import PageFactory.Guru99HomePage;

import PageFactory.Guru99Login;

public class Test99GuruLoginWithPageFactory {

    String driverPath = "C:\\geckodriver.exe";

    WebDriver driver;

```

```
Guru99Login objLogin;
```

```
Guru99HomePage objHomePage;
```

```
@BeforeTest
```

```
public void setup(){
```

```
    System.setProperty("webdriver.gecko.driver", driverPath);
```

```
    driver = new FirefoxDriver();
```

```
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

```
    driver.get("https://demo.guru99.com/V4/");
```

```
}
```

```
/**
```

```
 * This test go to https://demo.guru99.com/V4/
```

```
 * Verify login page title as guru99 bank
```

```
 * Login to application
```

```
 * Verify the home page using Dashboard message
```

```
 */
```

```

@Test(priority=0)

public void test_Home_Page_Appear_Correct(){

    //Create Login Page object

    objLogin = new Guru99Login(driver);

    //Verify login page title

    String loginPageTitle = objLogin.getLoginTitle();

    Assert.assertTrue(loginPageTitle.toLowerCase().contains("guru99 bank"));

    //login to application

    objLogin.loginToGuru99("mgr123", "mgr!23");

    // go the next page

    objHomePage = new Guru99HomePage(driver);

    //Verify home page

    Assert.assertTrue(objHomePage.getHomePageDashboardUserName().toLowerCase().contains("manger id : mgr123"));

}

}

```

La estructura completa del proyecto se verá como el diagrama:

Complete Project structure will
be like that

