Introducción a las pruebas automatizadas con Selenium-Java

Denis David Herrera Rojas

¿Qué es Maven?

Es una potente herramienta de gestión de proyectos que se utiliza para gestión de dependencias, como herramienta de compilación e incluso como herramienta de documentación. Es de código abierto y gratuita.

“Si no usamos Maven, tenemos que actualizar manualmente todas las librerías que usamos. Eso es un arduo trabajo de mantenimiento.”

Manos a la obra!

Crear nuestro primer proyecto Maven

Descargar las dependencias de

https://mvnrepository.com/

- Selenium java ->
- WebDriverManager ->
- J-Unit ->
- Test ng ->

¿Qué es Selenium?

Es una herramienta de código abierto para la automatización de pruebas,de navegadores web.

- Selenium IDE
- Selenium RC
- Selenium Webdriver
- Selenium Grid

Configuración de Webdriver

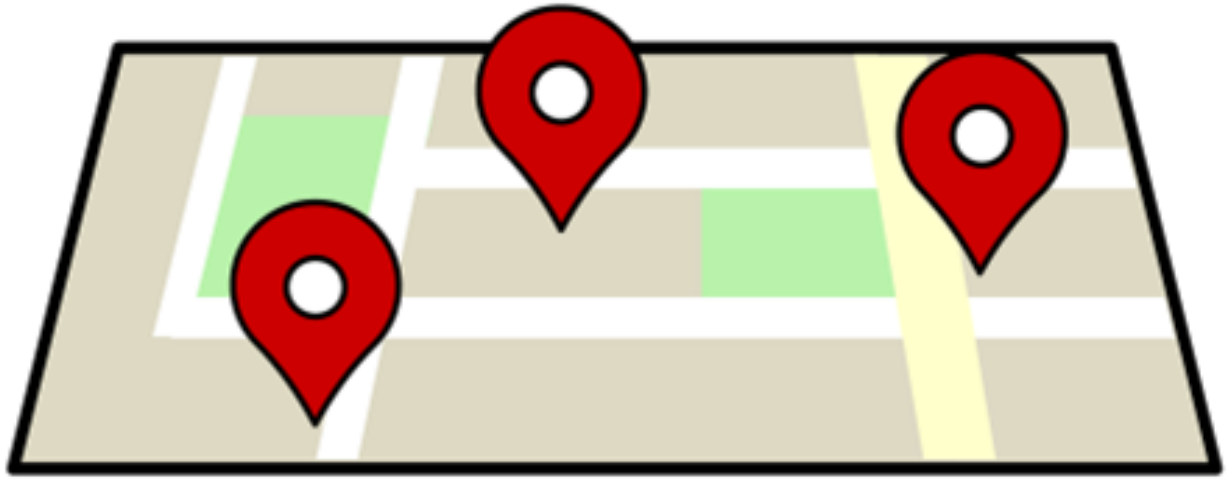
```
WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

driver.get("https://demo.guru99.com/test/login.html")
;
```

Localizadores

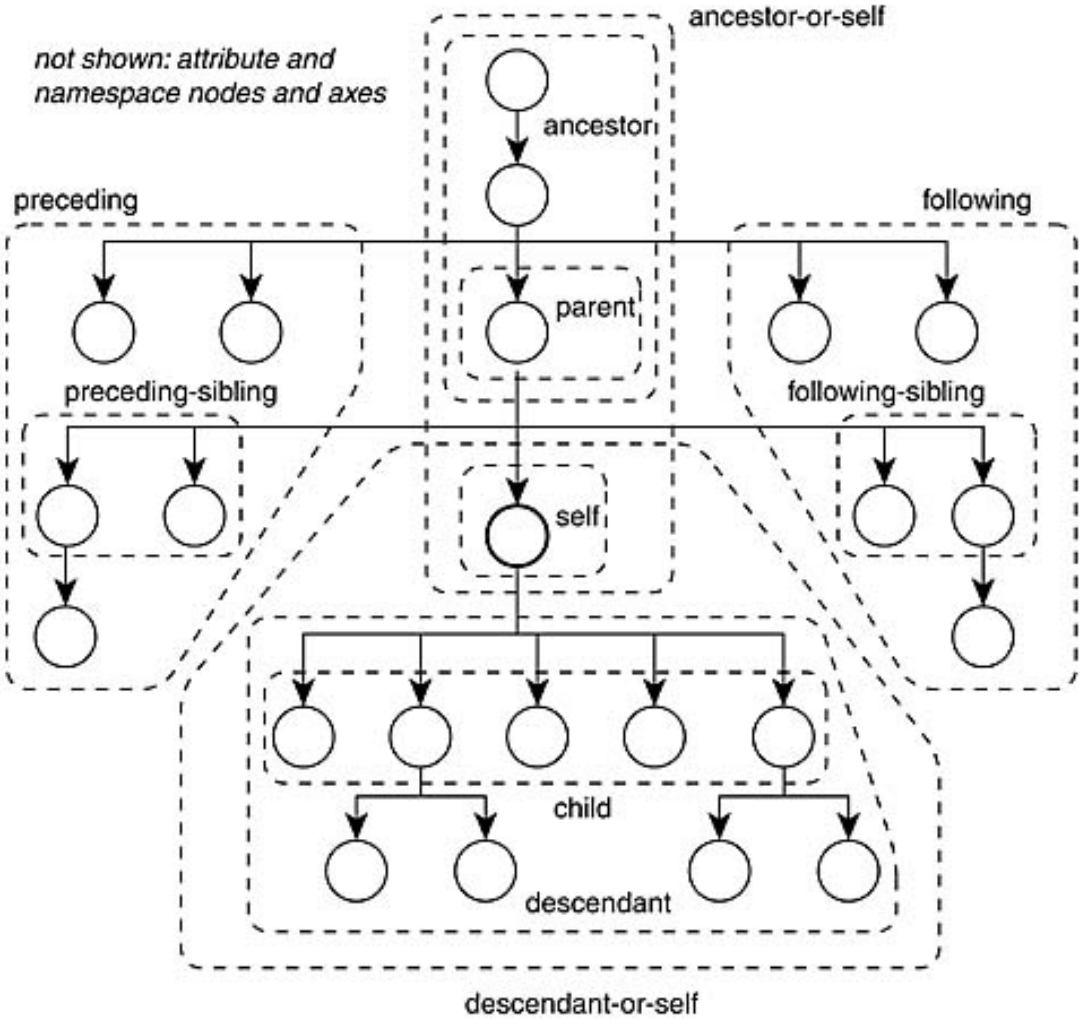
- By.id
- By.name
- By.className
- By.tagName
- By.linkText
- By.partialLinkText
- By.cssSelector
- By.xpath



¿Por qué es importante usar Xpath?

relativo -> /

absoluto -> //



Principales acciones de Selenium

get() -> Comando para abrir una URL específica

getCurrentUrl() -> Obtener la URL actual, nos sirve para verificar si la Url es correcta

findElement() -> Busca el primer elemento de la página actual que coincida con el localizador indicado

click() -> Hace click en un elemento enviado como parámetro

sendKeys() -> Escribe un valor a un elemento de tipo input enviado como parámetro

close() -> Cierra la pestaña actual del navegador

quit() -> Finaliza el webDriver

Para mas aciones revisar https://www.selenium.dev/documentation/webdriver/

Practiquemos -> https://demo.guru99.com/test/login.html

Implicit wait

“Las esperas implícitas se utilizan para proporcionar un tiempo de espera predeterminado (digamos 30 segundos) entre cada paso de prueba / comando consecutivo en todo el script de prueba”

```
WebDriver driver = new FirefoxDriver();
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
driver.get("http://somedomain/url_that_delays_loading");
WebElement myDynamicElement = driver.findElement(By.id("myDynamicElement"));
```

Explicit wait

“Las esperas explícitas se utilizan para detener la ejecución hasta que se cumpla una condición particular o haya transcurrido el tiempo máximo. A diferencia de las esperas implícitas, las esperas explícitas se aplican solo a una instancia en particular.”

```
WebDriver driver = new ChromeDriver();
driver.get("https://google.com/ncr");
driver.findElement(By.name("q")).sendKeys("cheese" + Keys.ENTER);
// Initialize and wait till element(link) became clickable - timeout in 10 seconds
WebElement firstResult = new WebDriverWait(driver, Duration.ofSeconds(10))
        .until(ExpectedConditions.elementToBeClickable(By.xpath("//a/h3")));
```

Algunas condiciones más usadas

elementToBeClickable - La condición esperada espera a que se pueda hacer clic en un elemento, es decir, debe estar presente / mostrado / visible en la pantalla y habilitado. *ExpectedConditions.elementToBeClickable ();*

textToBePresentInElement - La condición esperada espera a que un elemento tenga un determinado patrón de cadena.

ExpectedConditions.textToBePresentInElement ();

alertIsPresent - La condición esperada espera a que aparezca un cuadro de alerta. *wait.until*

ExpectedConditions.alertIsPresent ())! = null;

titleIs - La condición esperada espera una página con un título específico. *ExpectedConditions.titleIs ();*

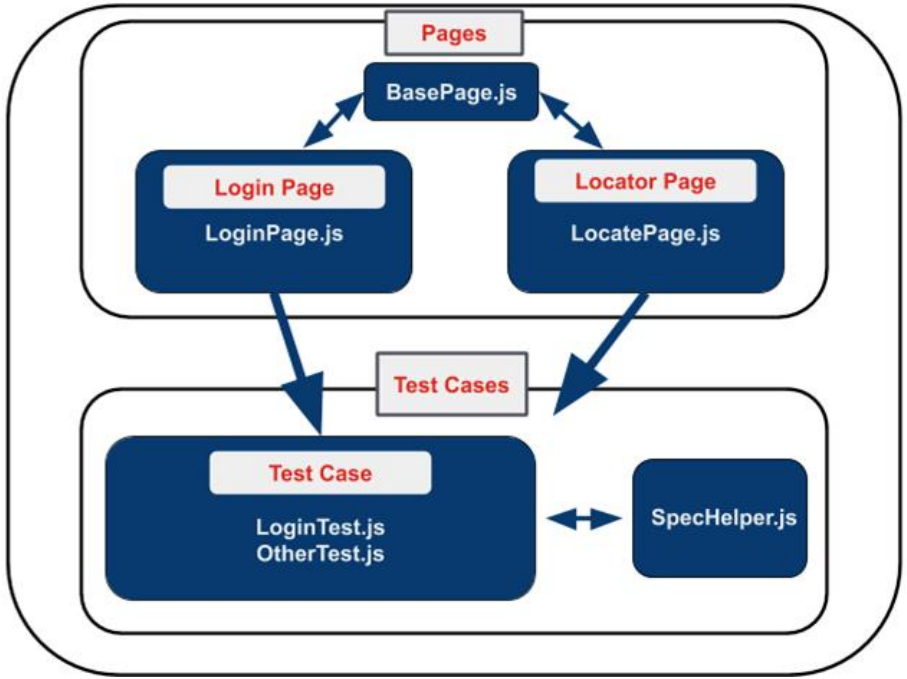
frameToBeAvailableAndSwitchToIt - La condición esperada espera a que un marco esté disponible y luego, tan pronto como el marco está disponible, el control cambia a él *ExpectedConditions.frameToBeAvailableAndSwitchToIt ();*

https://demoqa.com/dynamic-properties

Page Object Model

El matenimiento es mas practico

- Clase base -> Encapsula los métodos de selenium
- Clase Page -> Las acciones propias de una pagina y los localizadores
- Test -> Conjunto de acciones que verifican y validan un caso de prueba



Clase base

```java
public class Base {
    private WebDriver driver;

    public Base(WebDriver driver){
        this.driver = driver;
    }

    public WebDriver chromeDriverConnection(){
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        return driver;
    }

    public void openUrl(String url){
        driver.get(url);
    }

    public void closeBrowser(){
        driver.quit();
    }
```

```java
public class Localizadores {
    public static final By EMAIL_INPUT = By.id("email");
    public static final By PASSWORD_INPUT = By.id("passwd");
    public static final By BUTTON_LOGIN = By.name("SubmitLogin");
    public static final By WELCOME_MESSAGE = By.xpath("//h3[contains(text(),'Successfully Logged in...')]");
```

```java
public class LoginPage extends Base {
    public LoginPage(WebDriver driver) { super(driver); }

    public void login() {
        super.writeOnInput(Localizadores.EMAIL_INPUT, text: "email@emil.com");
        super.writeOnInput(Localizadores.PASSWORD_INPUT, text: "Password!");
        super.clickOnButton(Localizadores.BUTTON_LOGIN);
    }

    public void verifyLoginMessage() {
        if (!super.verifyIfElementIsDisplayed(Localizadores.WELCOME_MESSAGE)) {
            Assert.fail("Login message is not displayed");
        }

    }
}
```

Clase Page

Test

```
public class LoginTest {
    private WebDriver driver;
    LoginPage login;

    public LoginTest() {
    }

    @Before
    public void setUp() {
        this.login = new LoginPage(this.driver);
        this.driver = this.login.chromeDriverConnection();
        this.login.openUrl("https://demo.guru99.com/test/login.html");
    }

    @After
    public void tearDown() { this.login.closeBrowser(); }

    @Test
    public void test() {
        this.login.login();
        this.login.verifyLoginMessage();
    }
}
```

Test Suite

```
@RunWith(Suite.class)
@Suite.SuiteClasses({LoginTest.class, GoogleTest.class})
public class TestSuite {
    public TestSuite() {
    }
}
```

Practiquemos

Ingresar a: https://petstore.octoperf.com/actions/Catalog.action

- Registrar un usuario
- Iniciar sesión con el usuario registrado
- Verificar si el precio del perico del amazonas es $193.50