|  |  |
| --- | --- |
| C:\Users\63445142\Desktop\CURSO DEL CES AUTOMATIZACIÖN\Módulo 3\tarea5.png | Descripción breve  Se realiza el informe de la automatización de 4 escenarios del sistema  Ariagna Albuerne  Taller de automatización del Testing Funcional |

Contenido

[INTRODUCCIÓN 2](#_Toc79660957)

[DESARROLLO 3](#_Toc79660958)

[Configuración POM Del proyecto 5](#_Toc79660959)

[Parte B1 6](#_Toc79660960)

[Indicaciones para ejecución con clave y contraseña. 6](#_Toc79660961)

[8](#_Toc79660962)

[Herramientas Utilizadas 8](#_Toc79660963)

[CONCLUSIONES 9](#_Toc79660964)

# INTRODUCCIÓN

En el informe se muestra los elementos que se tuvieron en cuenta para darle solución a la consigna propuesta **‘**Implementar los scripts de prueba de la aplicación eGroupWare para los siguientes escenarios:

* Dar de alta un proyecto
* Consultar los datos de un proyecto
* Asignar recursos a un proyecto
* Agregar contactos a la agenda

# DESARROLLO

1. Dar de alta un proyecto

|  |
| --- |
| Guión   1. **IniciarSesion ()** 2. Clic en el menù project Manager 3. Clic en el botón **(**Add a new project) 4. Verificar que estamos en la ventana Egroupware [project Manager - Add project] 5. Ingresar título al proyecto “Proyecto de prueba” 6. GenerarIdProyecto() 7. Ingresar id del proyecto 8. Clic en Save 9. Verificar el mensaje “Project saved” 10. Verificar el mensaje Project saved 11. BuscarProyectoCreado() 12. EliminarProyectoCreado() 13. Verificar que se haya eliminado 14. Cerrar la pestaña Project Manager |

1. Consultar los datos de un proyecto

|  |
| --- |
| Guión   1. **IniciarSesion ()** 2. CrearProyecto() 3. BuscarProyectoCreado() 4. Verificar que todos los datos coincidan con el proyecto creado para consulta 5. EliminarProyectoCreado() 6. Cerrar la pestaña Project Manager |

1. Asignar recursos a un proyecto

|  |
| --- |
| Guión   1. **IniciarSesion ()** 2. CrearProyecto() 3. BuscarProyectoCreado() 4. Clic derecho en el proyecto 5. Seleccionar la opción open 6. Verificar que estamos en la ventana de edición 7. Ir hasta miembros y agregar un <miembro > 8. Clic en el botón sabe 9. BuscarProyectoCreado() 10. Verificar que se le haya agregado el recurso miembro 11. EliminarProyectoCreado() 12. Cerrar la pestaña Project Manager |

1. Agregar contactos a la agenda

|  |
| --- |
| **Guion:**   1. **IniciarSesion ()** 2. En la barra lateral clic en el menù Address Book 3. Clic en el botón (Add a new contact) 4. Verificar que estamos en la ventana Egroupware [Address Book] 5. Escribir datos nombre <nombre>; <prefix> 6. Clic en el botón ok 7. Escribir <movil>;<mail;><prefix>;<bussPhones>;<job>;<organizations>;<pais;> 8. Clic en save 9. Verificar el mensaje Contact saved 10. **BuscarContacto ()** 11. En el resultado verificar >;<movil>;<mail;><prefix>;<bussPhones>;<job>;<organizations>;<pais;> 12. EliminarContacto() 13. Cerrar la pestaña **Address Book** |

**Estructura del proyecto**

|  |
| --- |
|  |

|  |
| --- |
| Se realize una clase Base donde se inicializa el Webdriver una única vez así como objetos de selenium WebDriver, se crean métodos genéricos de acciones para emplear en las clases que extienden de ella.  package pages**;**  **import** org**.**openqa**.**selenium**.**JavascriptExecutor**;**  **import** org**.**openqa**.**selenium**.**WebDriver**;**  **import** org**.**openqa**.**selenium**.**By**;**  **import** org**.**openqa**.**selenium**.**WebElement**;**  **import** org**.**openqa**.**selenium**.**chrome**.**ChromeDriver**;**  **import** org**.**openqa**.**selenium**.**chrome**.**ChromeOptions**;**  **import** org**.**openqa**.**selenium**.**interactions**.**Actions**;**  **import** org**.**openqa**.**selenium**.**support**.**ui**.**ExpectedConditions**;**  **import** org**.**openqa**.**selenium**.**support**.**ui**.**WebDriverWait**;**  **import** org**.**openqa**.**selenium**.**support**.**ui**.**Select**;**  **import** java**.**util**.**Set**;**  **import** org**.**openqa**.**selenium**.**Keys**;**  public class BasePage **{**  protected static WebDriver driver**;**  private static WebDriverWait wait**;**  private static By messageSave**;**  static**{**  System**.**setProperty**(** "webdriver.chrome.driver" **,** "./src/test/resources/chromedriver.exe" **);**  ChromeOptions chromeOptions **=** **new** ChromeOptions**();**  driver **=** **new** ChromeDriver**(** chromeOptions **);**  wait **=** **new** WebDriverWait**(** driver **,** 20 **);**  messageSave **=** By**.**id**(** "egw\_message" **);**  **}**  public BasePage **(** WebDriver driver **){**  BasePage**.**driver **=** driver**;**  wait **=** **new** WebDriverWait**(** driver **,** 20 **);**  **}**  public static void navigateTo **(** String url **){**  driver**.**get**(** url **);**  **}**  public static void closedBrowser **(){**  driver**.**quit**();**  **}**  private WebElement find **(** By locator **){**  WebElement elementWithWait **=** wait**.**until**(** ExpectedConditions**.**visibilityOfElementLocated**(** locator **)** **);**  **return** elementWithWait**;**  **}**  public void write **(** By locator **,** String textToWrite **){**  find**(** locator **).**clear**();**  find**(** locator **).**sendKeys**(** textToWrite **);**  **}**  public void click **(** By locator **){**  find**(**locator**).**click**();**  **}**  public String getText **(** By locator **){**  **return** find**(** locator **).**getText**();**  **}**  public void clickJs **(** By locator **){**  WebElement waitUntil **=** wait**.**until**(** ExpectedConditions**.**visibilityOfElementLocated**(** locator **)** **);**  JavascriptExecutor js **=** **(**JavascriptExecutor**)** driver**;**  js**.**executeScript**(** "arguments[0].click();" **,** waitUntil **);**  **}**  public String winHandles **(** int positionWindow **){**  Set**<**String**>** windows **=** driver**.**getWindowHandles**();**  **return** **(**String**)** windows**.**toArray**()[** positionWindow **];**  **}**  public String getTitle **(){**  **return** driver**.**getTitle**();**  **}**  public void switchToVentana **(** String IdVentana **){**  driver**.**switchTo**().**window**(** IdVentana **);**  **}**  public boolean elementIsDisplayed **(** By locator **){**  **return** find**(** locator **).**isDisplayed**();**  **}**  public void enter **(** By selector **){**  **new** WebDriverWait**(** driver **,** 20 **).**until**(** ExpectedConditions**.**elementToBeClickable**(** selector **)** **).**click**();**  find**(** selector **).**sendKeys**(** Keys**.**ENTER **);**  **}**  public void clicRigthandMove **(** By locator **,** By locatorMove **){**  WebElement b **=** find**(** locator **);**  Actions action **=** **new** Actions**(** driver **);**  action**.**contextClick**(** b **).**build**().**perform**();**  WebElement move **=** find**(** locatorMove **);**  action**.**moveToElement**(** move **);**  move**.**click**();**  **}**  public String messageGetText **(){**  String mensaje **=** find**(** messageSave **).**getText**();**  click**(** messageSave **);**  **return** mensaje**;**  **}**  public void dobleClick **(** By locator **){**  Actions action **=** **new** Actions**(** driver **);**  WebElement doblec **=** find**(** locator **);**  action**.**doubleClick**(** doblec **).**build**().**perform**();**  **}**  public void selectOption **(** By locator **,** String value **){**  Select lista **=** **new** Select**(** find**(** locator **)** **);**  lista**.**selectByValue**(** value **);**  **}**  **}** |

|  |
| --- |
| Se creó clases para realizar los métodos sobre los elementos web de cada clase:  ProjectManagerPage  AddProjectPage  AddressBookPage  InicioPage  Se agregó un archivo properties donde se carga las credenciales y la url.  En el paquete tests  Se crearon una clase para los test relacionados con proyecto y otra para los test de los contactos.  TestAddressBook  TestProject  Luego existe una clase útil para la lectura de properties.txt |

### Configuración POM Del proyecto

Se realiza la configuración del POM para definir las dependencias a utilizarse:

|  |
| --- |
| <?xml version=**"1.0"** encoding=**"UTF-8"**?>  <project xmlns=**"http://maven.apache.org/POM/4.0.0"**  xmlns:xsi=**"http://www.w3.org/2001/XMLSchema-instance"**  xsi:schemaLocation=**"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  <modelVersion>**4.0.0**</modelVersion>  <groupId>**tarea4**</groupId>  <artifactId>**tarea4**</artifactId>  <version>**1.0-SNAPSHOT**</version>  <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->  <dependencies>  <dependency>  <groupId>**org.seleniumhq.selenium**</groupId>  <artifactId>**selenium-java**</artifactId>  <version>**3.141.59**</version>  </dependency>  <dependency>  <groupId>**org.seleniumhq.selenium**</groupId>  <artifactId>**selenium-chrome-driver**</artifactId>  <version>**3.141.59**</version>  </dependency>  <dependency>  <groupId>**org.junit.jupiter**</groupId>  <artifactId>**junit-jupiter-engine**</artifactId>  <version>**5.8.0-M1**</version>  <scope>**test**</scope>  </dependency>  </dependencies>  </project> |

Estructura de las clases Test Paso el ejemplo de la de TestAddressBook

|  |
| --- |
|  |

## Parte B1

* Automatizar con Selenium WebDriver el siguiente guion

1. Acceder a la URL "capacitacion.ces.com.uy"
2. Iniciarse sesión con un usuario y contraseña existentes y habilitados
3. Acceder al curso "Taller de Automatización del Testing Funcional" (que se encuentra en la sección "Mis Cursos")
4. En la sección "Buscar en los foros", buscar un foro que contenga la palabra "Bienvenida"

## Indicaciones para ejecución con clave y contraseña.

Se entrega un fichero leeme.txt con las indicaciones.

Estructura del proyecto

|  |
| --- |
|  |

Solución

|  |
| --- |
| package cesAutomation**;**  **import** org**.**junit**.**jupiter**.**api**.**AfterAll**;**  **import** org**.**junit**.**jupiter**.**api**.**BeforeAll**;**  **import** org**.**junit**.**jupiter**.**api**.**Test**;**  **import** org**.**junit**.**jupiter**.**api**.**TestInstance**;**  **import** org**.**openqa**.**selenium**.**By**;**  **import** org**.**openqa**.**selenium**.**WebDriver**;**  **import** org**.**openqa**.**selenium**.**WebElement**;**  **import** org**.**openqa**.**selenium**.**chrome**.**ChromeDriver**;**  **import** java**.**util**.**concurrent**.**TimeUnit**;**  **import** static org**.**junit**.**jupiter**.**api**.**Assertions**.**assertEquals**;**  **import** java**.**util**.**ArrayList**;**  **import** util**.**LeeFichero**;**  @TestInstance**(**TestInstance**.**Lifecycle**.**PER\_CLASS**)**  public class CesAutomation **{**  private WebDriver driver**;**  By link **=** By**.**cssSelector**(** ".g:nth-child(1) > div:nth-child(2) .LC20lb" **);**  By linkAcceder **=** By**.**linkText**(** "Acceder" **);**  By nombre **=** By**.**id**(** "username" **);**  By password **=** By**.**id**(** "password" **);**  By entrar **=** By**.**id**(** "loginbtn" **);**  By paginaCursos **=** By**.**linkText**(** "INTRODUCCION A LA AUTOMATIZACION DEL TESTING FUNCIONAL 202103" **);**  By foro **=** By**.**cssSelector**(** "#module-21581 .instancename" **);**  By linkBienvenida **=** By**.**linkText**(** "Bienvenida" **);**  @BeforeAll  public void setUp **(){**  System**.**setProperty**(** "webdriver.chrome.driver" **,** "./src/test/resources/chromedriver.exe" **);**  driver **=** **new** ChromeDriver**();**  driver**.**manage**().**window**().**maximize**();**  **}**  @Test  public void cesBusqueda **(){**  LeeFichero credencial**=** **new** LeeFichero**();**  ArrayList**<**String**>** credenciales**=**credencial**.**leerFichero**();**  String user**=**credenciales**.**get**(** 0 **);**  String contra**=**credenciales**.**get**(**1**);**  driver**.**get**(** "https://capacitacion.ces.com.uy" **);**  driver**.**manage**().**timeouts**().**implicitlyWait**(** 5 **,** TimeUnit**.**SECONDS **);**  driver**.**findElement**(** linkAcceder **).**click**();**  WebElement username **=** driver**.**findElement**(** nombre **);**  username**.**clear**();**  username**.**sendKeys**(**user**);**  WebElement clave **=** driver**.**findElement**(** password **);**  clave**.**clear**();**  clave**.**sendKeys**(**contra **);**  WebElement acceder **=** driver**.**findElement**(** entrar **);**  acceder**.**submit**();**  WebElement pagCursos **=** driver**.**findElement**(** paginaCursos **);**  pagCursos**.**click**();**  WebElement novedades **=** driver**.**findElement**(** foro **);**  novedades**.**click**();**  assertEquals**(** "Novedades" **,** driver**.**getTitle**()** **);**  WebElement bienvenida **=** driver**.**findElement**(** linkBienvenida **);**  assertEquals**(** "Bienvenida" **,** bienvenida**.**getText**()** **);**  **}**  @AfterAll  public void tearDown**(){**  driver**.**quit**();**  **}**  **}** |

**Captura del test pasando y con la versión de Chrome utilizada.**

## 

## Herramientas Utilizadas

* IntelliJ IDEA Community Edition
* Java versión 1.8.0\_281
* Maven
* JUnit 5 (junit-jupiter-engine 5.8.0-M1)
* Selenium (3.141.59)

Parte B2

¿Es posible automatizar la descarga del archivo PDF "Material cubrimiento" del "Módulo 3" solamente utilizando Selenium? Justifique.

Entiendo que no es posible solamente con Selenium, hay que agregar configuraciones específicas dependencia del navegador, agregando configuración específica del navegador (en Firefox agregando Firefox Profile y seteando configuración específica…) Chrome parte de lo que podría hacerse es activar la descarga automática desde la configuración del navegador, pero no es lo único.

# CONCLUSIONES

Con la realización de este ejercicio se pudo comprobar que:

1. Es útil apoyarse de herramientas como selenium IDE para obtener los selectores.
2. Hay diferentes tipos de selectores
3. La importante la utilización de Junit 5 con Selenium