



DESCRIPCIÓN BREVE

Instructivo para la plataforma de automatización

[Ariagna Albuerne](#)

Taller de automatización del Testing Funcional

Contenido

INTRODUCCIÓN.....	2
DESARROLLO	3
Estructura del proyecto.....	3
Paquete pages	3
Paquete Suite.....	3
Paquete test	3
Paquete properties:	3
resources:	3
Estructura del proyecto (imagen)	3
Suite	4
Etiquetas	4
Gestión de datos de pruebas.....	4
Navegadores	5
Ejemplo: ejecutar los test desde IntelliJ	5
Desde la testSuite en el navegador Chrome	5
Herramientas Utilizadas	6

INTRODUCCIÓN

La plataforma de automatización del sistema egroupware Smart online office está desarrollada con el objetivo de comprobar algunos de los principales flujos de casos exitosos del sistema, en esta versión (versión 1.0) se trabajó sobre los módulos proyecto y contactos.

DESARROLLO

Estructura del proyecto

La plataforma es un proyecto maven cuya estructura es la siguiente:

Paquete pages: Cuenta con 5 clases, la clase base donde se definen todos los métodos para ejecutar acciones con los WebElement y 4 clases donde se definen los métodos específicos de navegación en las mismas.

Paquete Suite: Cuenta con una clase TestSuite donde se definen las clases test a ejecutar en este caso se agruparon los dos test de cada módulo, proyecto y agenda, pero son lo bastante flexible para que a medida que el proyecto crezca definir un criterio diferente para las testSuite.

Paquete test: cuenta con dos clases 1 para los test relacionados a proyecto y otro para los test relacionados a Agenda.

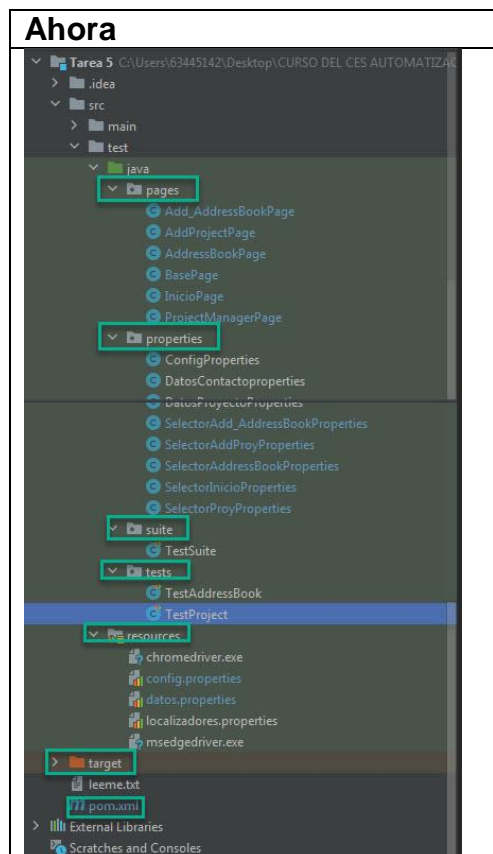
Paquete properties: contiene la clase métodos leer el archivo de datos, localizadores y config donde se define el navegador.

resources: contiene los driver de Chrome y Edge y los archivos .properties para datos y selección de navegadores.

Target: contienen los archivos de reportes generados cuando se corren los test usando maven (mvn test site)

Archivo pom.xml con las dependencias desde el repositorio de maven.

Estructura del proyecto (imagen)



Suite

- Se crea una clase TestSuite para ejecutar las dos clases test
`@SelectClasses({TestAddressBook.class,TestProject.class})`

Etiquetas

- Etiqueta `@DisplayName` a los test con nombres a mostrar cuando se ejecutan un poco más descriptivos.
- Etiqueta `@IncludeTags("Para_Suite_AmbienteTest")`
Para simular que en esta se pondría la etiqueta de las pruebas que corresponden a un ambiente determinado en este caso de test, por si luego se le agregan otros test para otros ambientes.
- Etiqueta `@TestMethodOrder(MethodOrderer.OrderAnnotation.class)` para definir orden dentro de los test de las clases test.
- Etiqueta `@Order()` para definir el orden de los test de la clase TestProyect para cambiar el orden definir el número ejemplo `@Order(1)`

Gestión de datos de pruebas

En resources se encuentra el archivo datos.properties en el cuál se puede gestionar los datos de pruebas.

- En el caso de las credenciales se deben cargar en user (usuario) y pass (contraseña)
Para el caso de prueba Crear proyecto, se puede cambiar el título del proyecto por el que se desee.
- En el caso de prueba de agregar recursos se puede cambiar el miembro que se le asigna al proyecto, siempre y cuando exista y cumpla con el formato <APELLIDO, NOMBRE>
- En el caso de prueba consultar proyecto pudieran cambiarse los datos pero la precondition es que debe existir un proyecto al que se le agregue datos correspondientes a las claves que existen en el archivo.
- En el caso de crear contacto, se pueden variar todos los datos

Ejecución

Puede ejecutarse desde la TestSuite, o desde cada paquete test las clases o métodos de forma independiente o empleado el comando `mvn test` o `mvn test site` (este último genera un reporte)

Navegadores

Se agregó un archivo config.properties en el que se puede definir el navegador, en este caso se hizo una lógica en BasePage para los navegadores Chrome y Edge. Para cambiar solo hay que ir a el archivo config y en la clave browser cambiar por valor browser=Chrome para ejecutar en Chrome y browser=edge para ejecutar en Edge.

Las versiones de los navegadores:

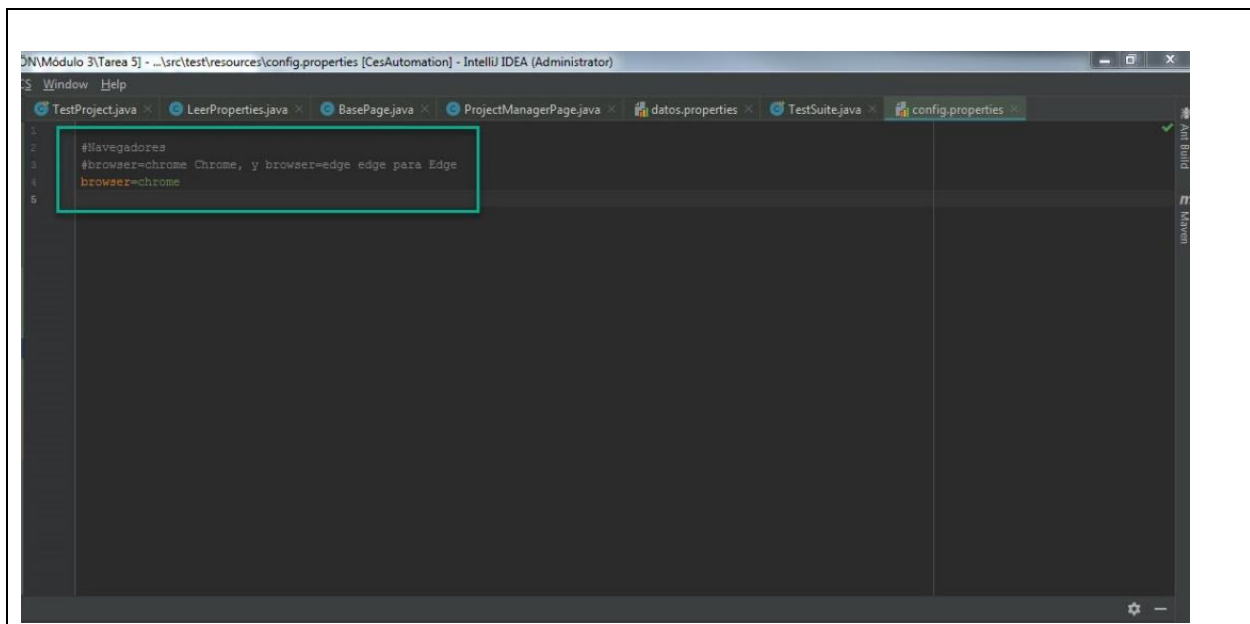
Chrome : Versión 92.0.4515.159 (Build oficial) (64 bits)

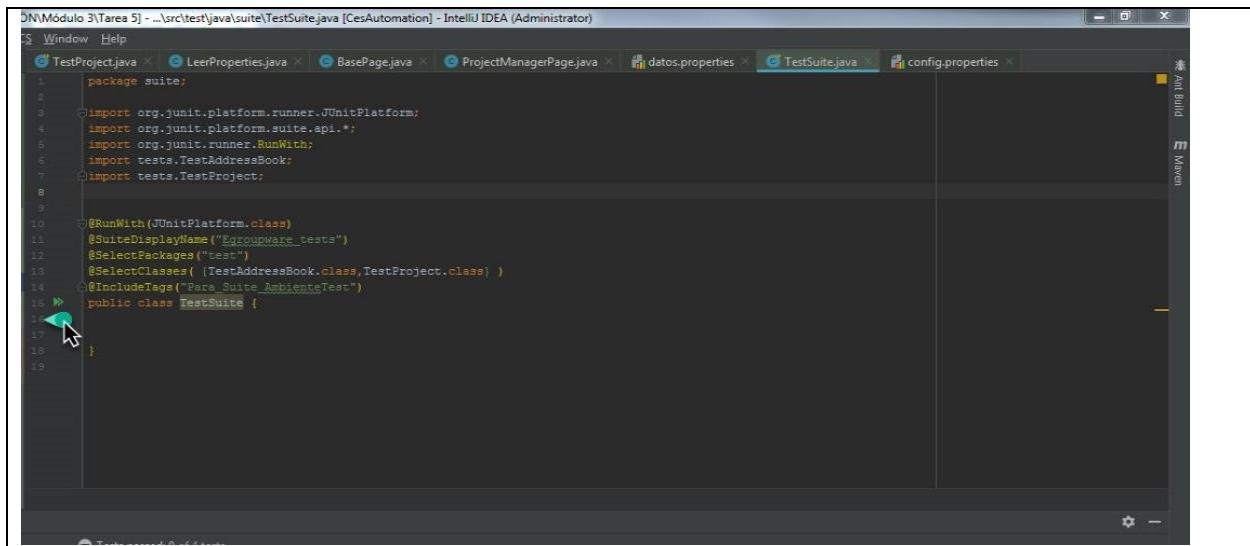
Microsoft Edge Versión 92.0.902.78 (Compilación oficial) (64 bits)

Ejemplo: ejecutar los test desde IntelliJ

Desde la testSuite en el navegador Chrome

- 1- Ir al archivo config.properties
- 2- A la clave browser asignarle el valor Chrome (browser=chrome)
- 3- Ir al archivo datos.properties y revisar y cambiar de requerir los datos de pruebas como más adelante se explica.
- 4- Ir la clase TestSuite clic derecho Run testSuite.





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)

Código completo se encuentra en git: <https://github.com/aalbuerner87/tareaCesSelenium>