



[DESCRIPCIÓN BREVE](#)

Se realiza el informe describiendo las pruebas y los cambios realizados

[Ariagna Albuerne](#)

Taller de automatización del Testing Funcional

Contenido

INTRODUCCIÓN	2
DESARROLLO	3
Correcciones	3
Descripción de las Pruebas	4
TestProject	4
TestAddressBook	4
Ejecución	5
• Inconvenientes y dificultades encontradas	6
Herramientas Utilizadas	6
Ventajas y desventajas de las herramientas utilizadas	6
CONCLUSIONES	7
Anexos.....	8

INTRODUCCIÓN

En el informe se muestra los elementos que se tuvieron en cuenta para darle solución a la consigna propuesta:

Consigna de la actividad

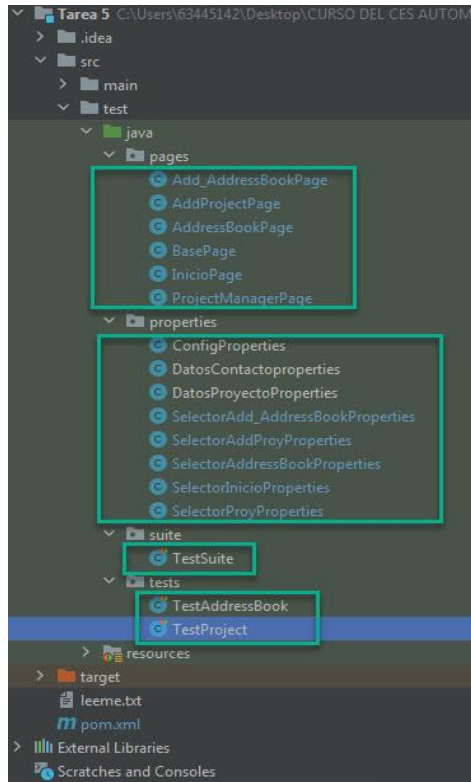
- Verificar los scripts y suites de la plataforma.
- Agregar y/o corregir lo que entienda necesario en base al feedback de actividades anteriores.
- Ejecutar las pruebas.

Además realizará un análisis para determinar:

- Ventajas y desventajas (conclusión posterior a su experiencia) de las herramientas utilizadas
- Inconvenientes y dificultades encontradas

DESARROLLO

Estructura Actual del proyecto

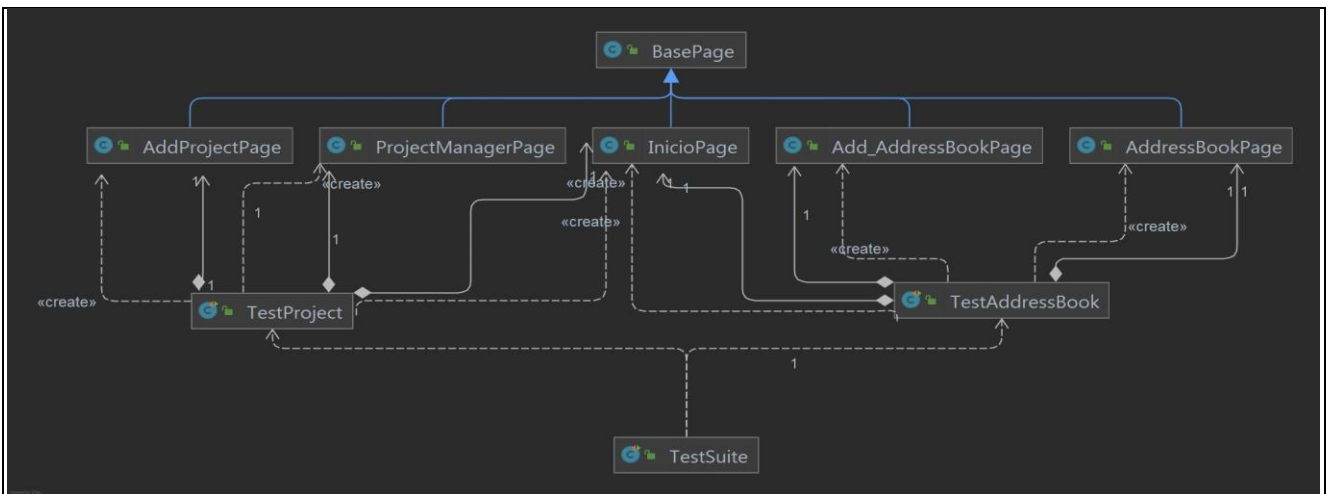


Se anexan diagramas de las relaciones de las clases, lo que hace que se entienda de forma clara la estructura del proyecto. [Ver anexo I](#)

Correcciones

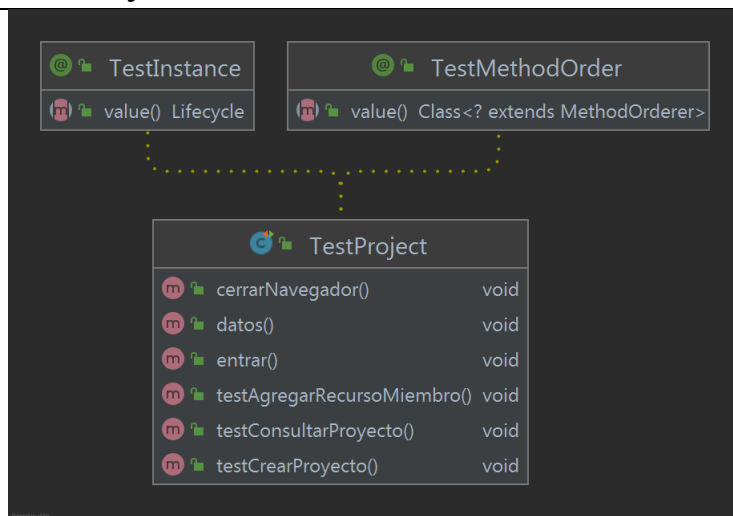
- Se agrega gestión de localizadores a través de los archivos .properties donde están los localizadores debidamente identificados de acuerdo a la página donde se utilizan y el tipo de selector que es.
 - Para la solución de esto se decide tener archivos .properties y clases para gestionarlas, por lo que en las clases page se obtiene el elemento abstrayéndose de la clave que se maneja en los archivos.
- Se agrega una clase para gestionar la página de agregar contacto
- Se hace un refactor del test testCrearContacto () realizando un método por cada acción por lo que se entiende los pasos del guion de forma clara.
- Se hace un refactor del test testConsultarProyecto () realizando un método para validar cada elemento que se obtiene y se cambia de assertEquals a assertTrue debido a que estos métodos devuelven un boolean.
- Se agrega en la clase TestProject la etiqueta @Order(num) para definir el orden de los test de esta clase.
- Se agregan dependencias maven para generar un sitio de reportes con reporte de los resultados de las pruebas y características generales del proyecto.
- Se arregla para que en contacto se pueda agregar cualquier país y que este sea modificado a través del archivo de gestión de datos .properties
- Se agrega que validación para que si el archivo config.properties no tiene la clave se ejecute por defecto en el navegador Chrome.

Descripción de las Pruebas



El proyecto cuenta con una testSuite y dos clases Test: TestProject y TestAddressBook

TestProject



testCrearProyecto ()

Este test agrega un nuevo proyecto, comprueba que se ha agregado correctamente, luego elimina el proyecto agregado.

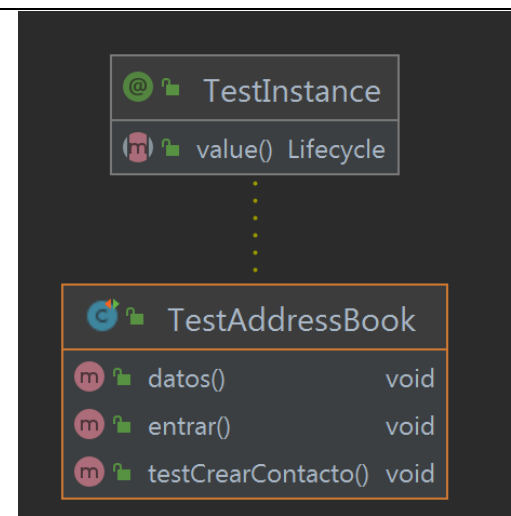
testConsultarProyecto ()

Este test consulta un proyecto existente creado, y valida los datos devueltos que coincidan con los datos que se encuentran en el archivo properties.

testAgregarRecursoMiembro ()

Este test se crea sus propios datos de prueba, crea un proyecto, lo busca lo edita, agregándole un nuevo recurso (miembro)

TestAddressBook



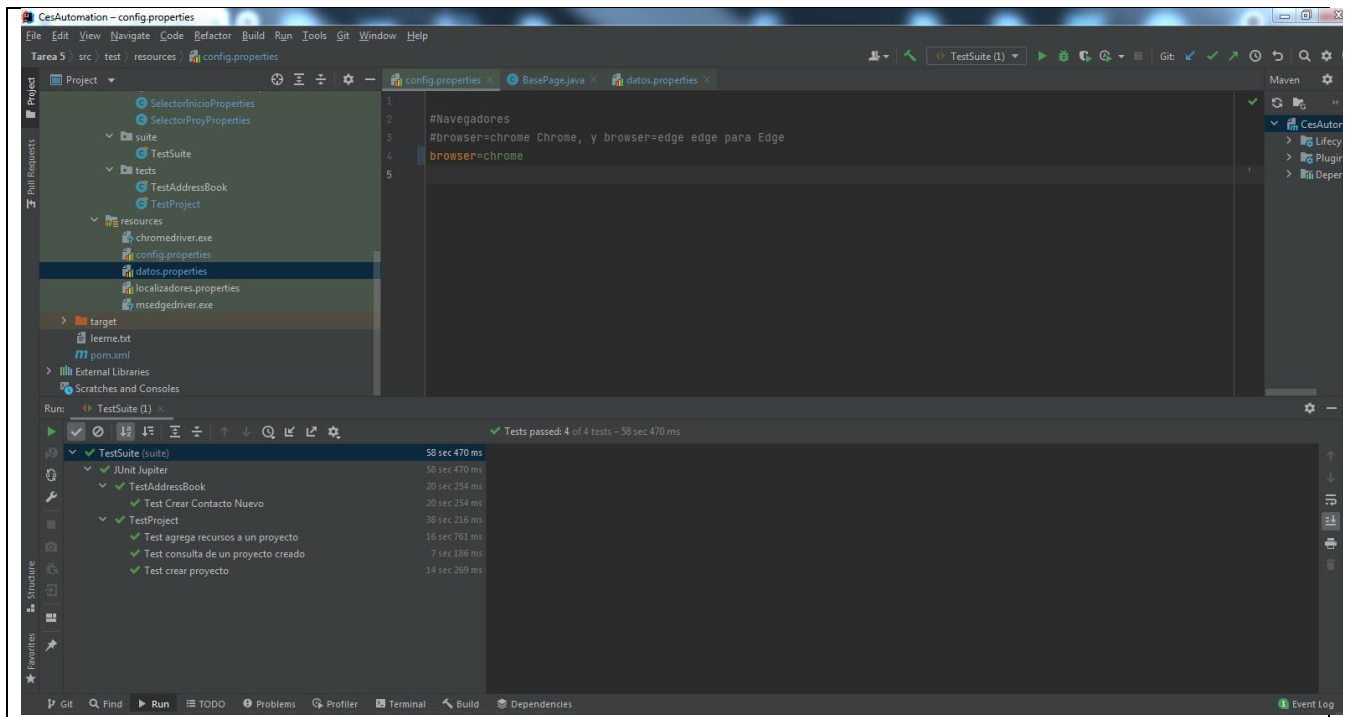
testCrearContacto ()

Este test crea un contacto con varios datos, agrega dato por dato, guarda el contacto, lo busca y valida que los elementos de la búsqueda coincidan con los agregados.

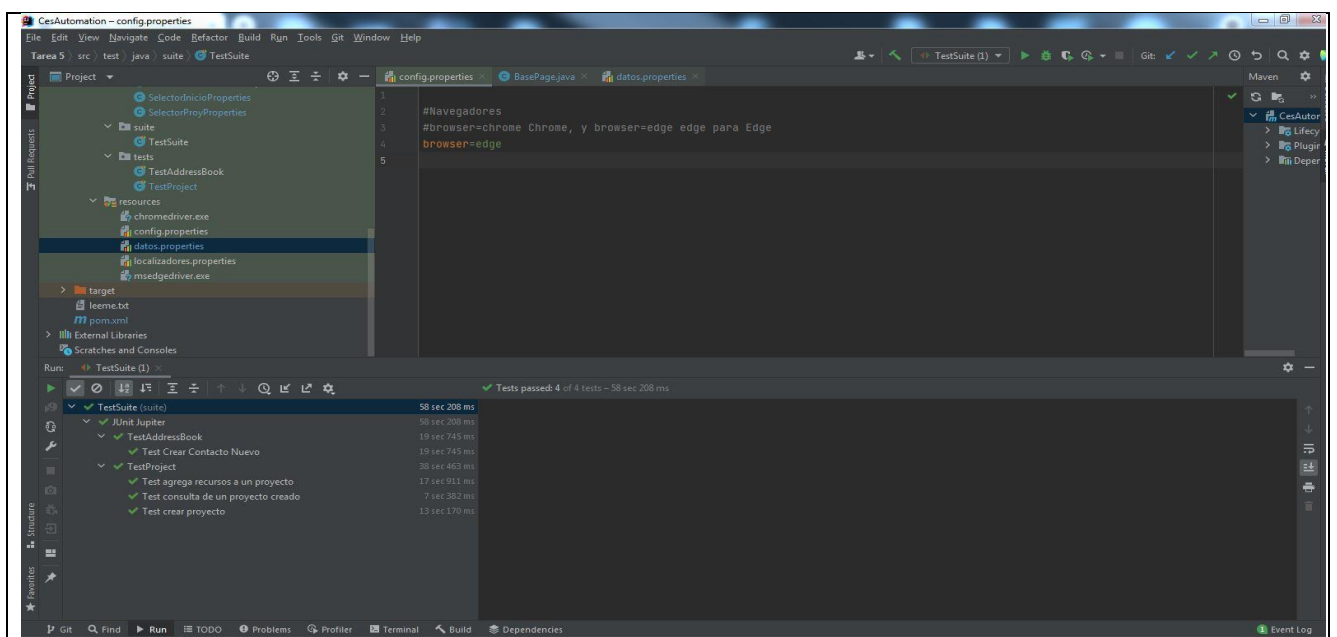
Ejecución

Los test puede ejecutarse en 2 navegadores desde la TestSuite, o desde cada paquete test las clases o métodos de forma independiente, y por el comando de maven 'mvn test' ([ver instructivo adjunto](#))

Captura del test pasando (nuevos cambios) Chrome.



Captura del test pasando (nuevos cambios) Edge



• Inconvenientes y dificultades encontradas

Los principales inconvenientes encontrados en todo el proceso de desarrollo del framework fue sin dudas encontrar los localizadores adecuados, ya que existían diferentes elementos web con el mismo localizador, localizadores dinámicos y fue trabajoso armarlos y fue necesario en la mayoría comprobar su unicidad empleando la herramientas de desarrollo de los propios navegadores.

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

Herramientas Utilizadas

- IntelliJ IDEA Community Edition
- Java versión 1.8.0_281
- Maven

Selenium			JUnit 5		
GroupId	ArtifactId	Version	GroupId	ArtifactId	Version
org.seleniumhq.selenium	selenium-chrome-driver	3.141.59	org.junit.jupiter	junit-jupiter-api	5.8.0-M1
org.seleniumhq.selenium	selenium-java	3.141.59	org.junit.jupiter	junit-jupiter-engine	5.8.0-M1
			org.junit.jupiter	junit-jupiter-params	5.7.0
			org.junit.platform	junit-platform-runner	1.7.2
			org.junit.platform	junit-platform-suite-api	1.8.0-M1

Ventajas y desventajas de las herramientas utilizadas

Como ventajas de emplear Junit 5 podemos mencionar que es Java por lo que el que esté familiarizado con el lenguaje es fácil de entender e implementar. De forma sencilla se pueden crear las Suite de pruebas, e incorpora etiquetas que facilitan el trabajo, ya sea de orden de ejecución e identificación de las pruebas. Se integra fácilmente con herramientas como maven que fue una de las que empleé en este framework.

El empleo de selenium Web Driver es esencial para manejar los elementos web de las páginas, proporcionando los métodos necesarios para interactuar con las páginas desde varios navegadores y es de fácil integración con diferentes lenguajes y framework, demostrándose en el desarrollo de este framework. Para el alcance de esta aplicación no resultó difícil pero es probable que para casos más complejos de esta aplicación con elementos web más complejos requiere un mayor tiempo de aprendizaje para emplearlo de forma más eficiente.

Podríamos mencionar como desventaja que al emplear un driver específico por navegador esto es bueno porque asegura que ejecute la prueba en el navegador adecuado pero a su vez es dependiente de este driver para ejecutar la prueba y que sea el correcto de acuerdo a la versión del navegador que se esté empleando.

CONCLUSIONES

Con la realización de este ejercicio se pudo comprobar que:

- Dentro del proceso de construcción del software la instancia de seleccionar los selectores adecuados es fundamental, entiendo que es mejor no apoyarse de herramientas de grabación y si se emplean comprobar luego el selector en el navegador.
- Crear el proyecto con una estructura clara, incluyendo nombre de variables y métodos facilita el proceso de entender el código y encontrar errores.
- El uso de archivos .properties, es fácil de emplear, es versátil porque puede emplearse tanto para manejo de datos, configuración y empleo de selectores de una manera más sencilla y fácil de entender.
- El uso de maven permite mejor versatilidad con el agregado y actualización de dependencias Junit y otras que necesites para el proyecto.
- Elegir el método assert adecuado dependiendo del caso mejora la calidad de las pruebas
- El empleo de las etiquetas de JUnit ayudan a inicializar, establecer orden y limpiar los datos de prueba facilitan la realización de las mismas.

Anexos

Anexo I. Relación entre las clases del proyecto

