

Título

Manual del Usuario

Breve Descripción

Manual de usuario de Selenium

Propósito

La creación de este manual se realiza con el propósito de dar a conocer a detalle las acciones requeridas para la automatización de pruebas funcionales.

Elaborado por

SAXA

Última revisión

10/03/2021

Referencias

Versión

1.0

Válido

De: Diciembre/2020

A: Diciembre/2021

Sustituye al Documento

Estado

☐ ☐ Draft

☐ ☒ Definitive

Lista de Distribución

SC Manager	RGA
Scrum Master	COMS
Software Project Manager	COMS
Technical Leader	MORR
SCM Group	BEGI, HEOX
SQA Group	RURL, TOBF
Software Engineering Group	MORR, MALZ
Test Group	ANBM, SARE, RITL, LORO
Data Analyst	MROM

Autorización

Elaborado por

Revisado por

Revisado por /
Autorizado por

SAXA

1. INTRODUCCIÓN	3
1.1 SELENIUM	3
1 CONEXIÓN AL REPOSITORIO	4
2 EJECUCIÓN DE PRUEBAS AUTOMATIZADAS.	9
2.1 GENERACIÓN DE REPORTE DE PRUEBAS.....	12
2.2 CAPTURA DE EVIDENCIAS	15

1. INTRODUCCIÓN

En el presente documento se describe el manejo de Selenium, herramienta dedicada a la automatización de pruebas funcionales.

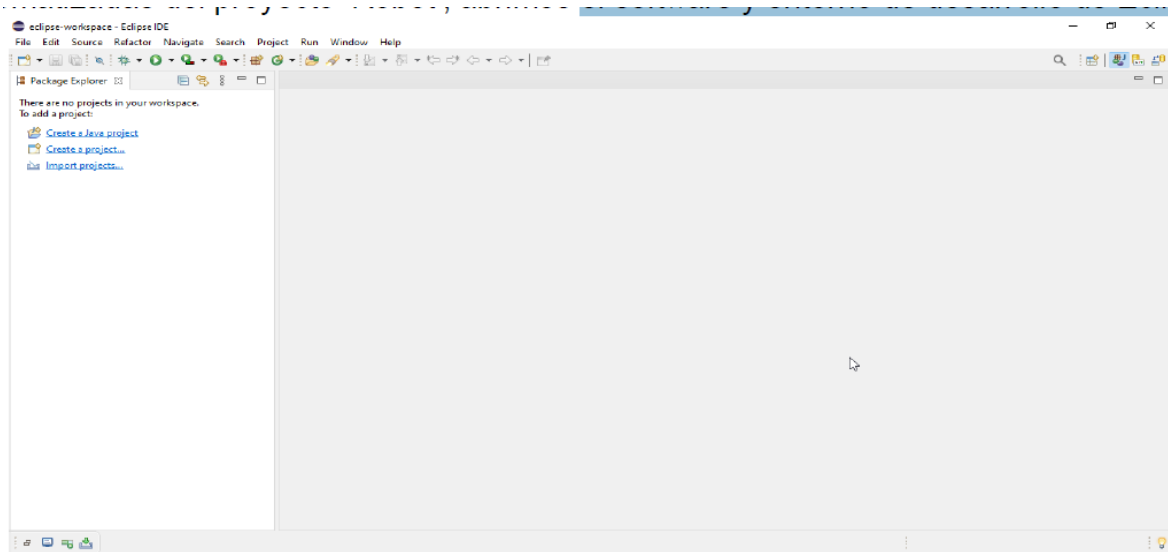
El objetivo de este manual es brindar a detalle los pasos que se requieren realizar para ejecutar las pruebas en Selenium a través del software TestNG, así como la generación de reportes, una vez que estas sean ejecutadas.

1.1 SELENIUM

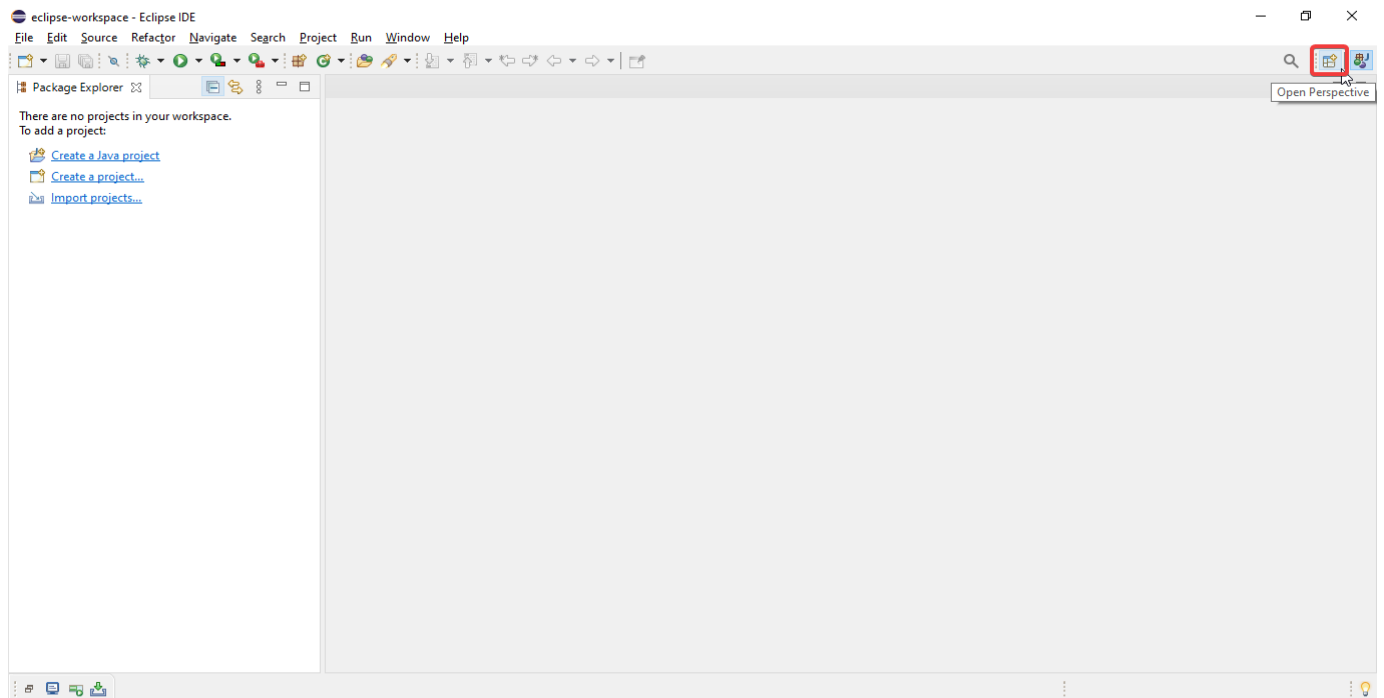
Selenium es un entorno de pruebas que se utiliza para comprobar si el software que se está desarrollando funciona correctamente. Incluye también un lenguaje específico de dominio para pruebas y de esta manera escribir pruebas en un amplio número de lenguajes de programación populares incluyendo Java, C#, Ruby, Groovy, Perl, Php y Python. Las pruebas pueden ejecutarse usando la mayoría de los navegadores web modernos en diferentes sistemas operativos como Windows, Linux y OSX.

1 Conexión al repositorio

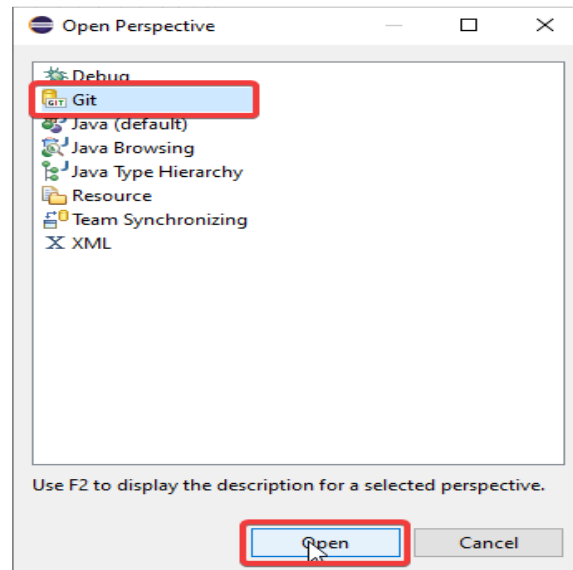
1. Abra el software y entorno de desarrollo de Eclipse.



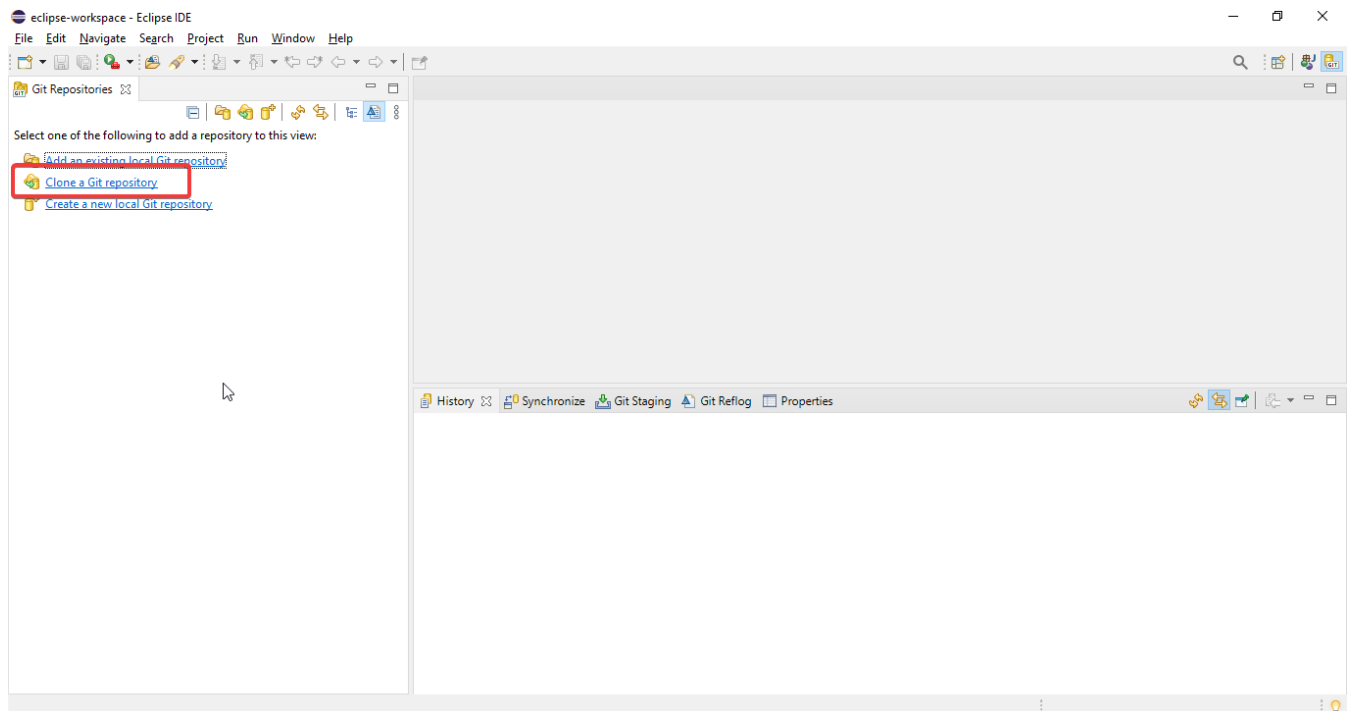
2. En la parte superior derecha, presione el botón "Open perspective".



3. En la ventana que se abre, seleccione la opción 'Git' y de clic en el botón "Open".



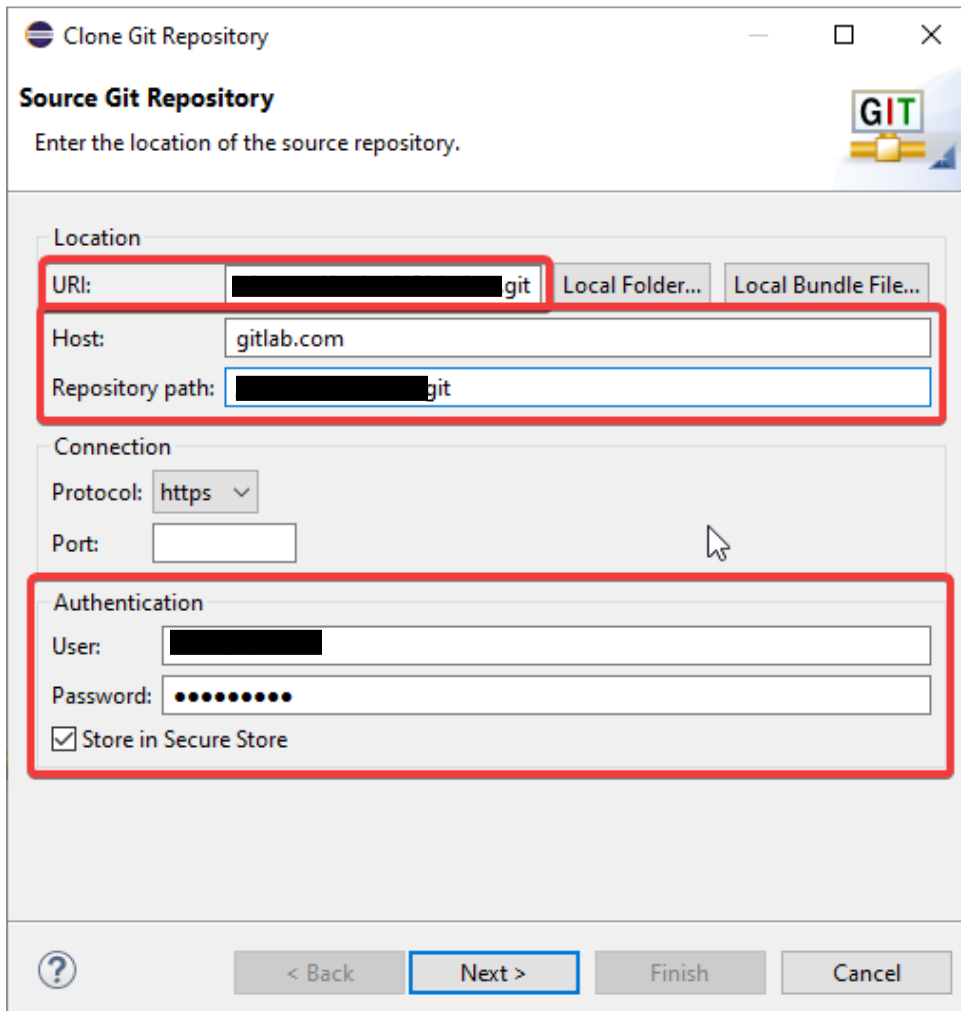
4. En la perspectiva de Git que se acaba de abrir, haga clic en la opción "Clone a Git repository".



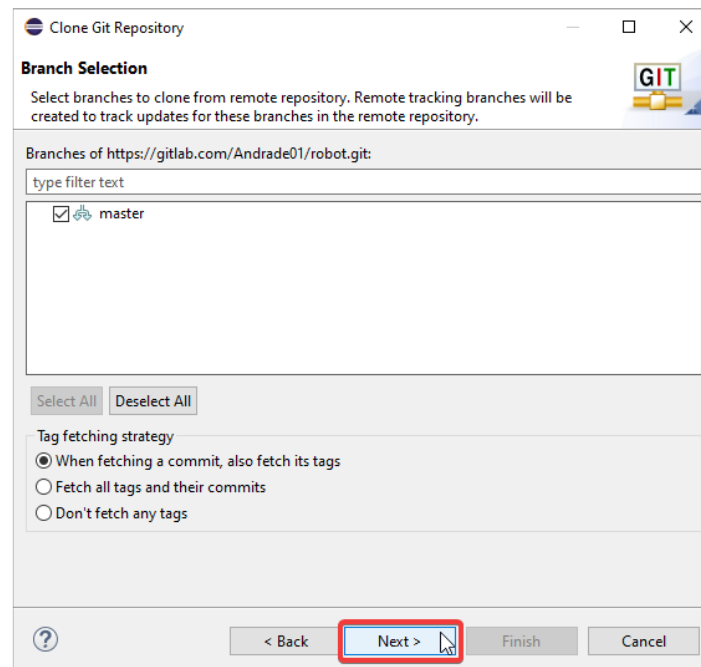
5. En la ventana que se abre, coloque los datos del repositorio del cual va a clonar su proyecto, de la siguiente manera:

- **URL** - Escriba la ruta para clonar su repositorio desde GitLab
- **Host y Repository path** – Son campos que se llenan al colocar el campo URL.
- La sección “**Connection**”, la dejamos tal cual se presenta.
- En la sección “**Authentication**” colocar usuario y contraseña que tiene asignado en GitLab.

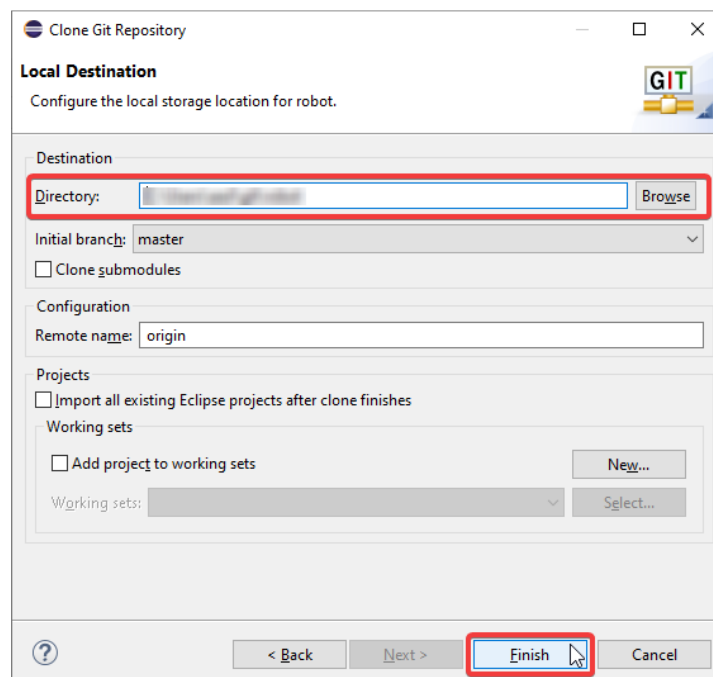
Al terminar de colocar los campos de clic en el botón ‘Next’.



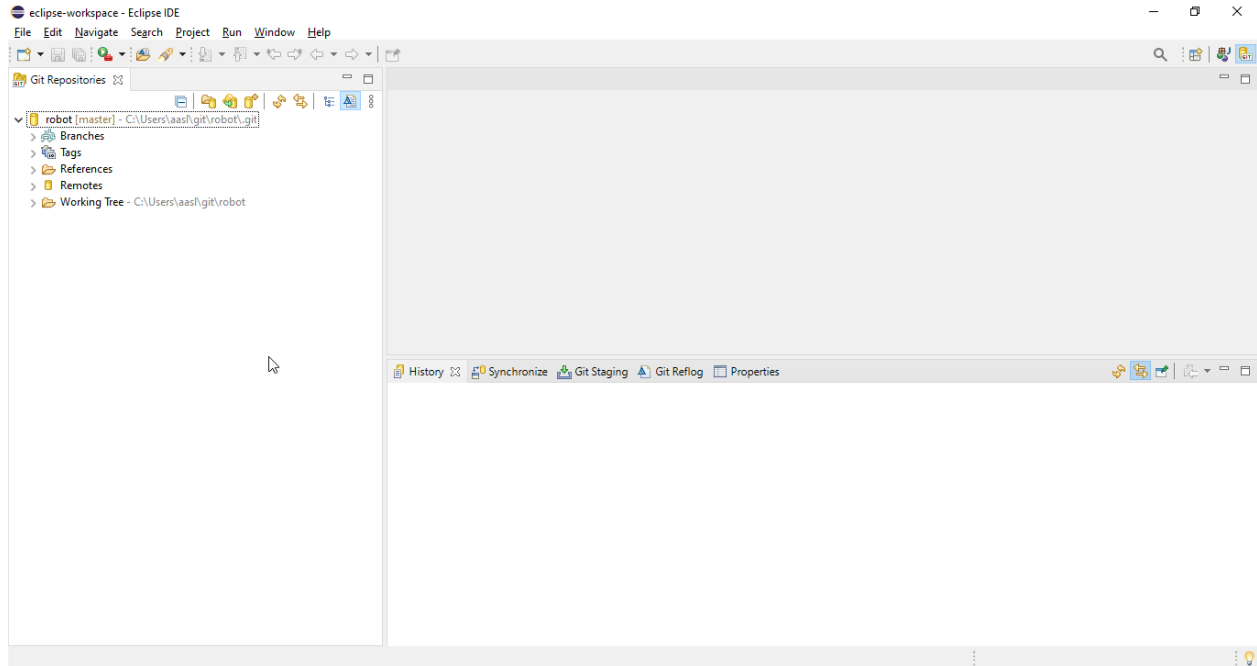
6. A continuación, haga clic en el botón 'Next'.



7. En el campo "Directory" escriba la ruta donde desea clonar su proyecto, una vez definida la ruta de clic en el botón 'Finish'.

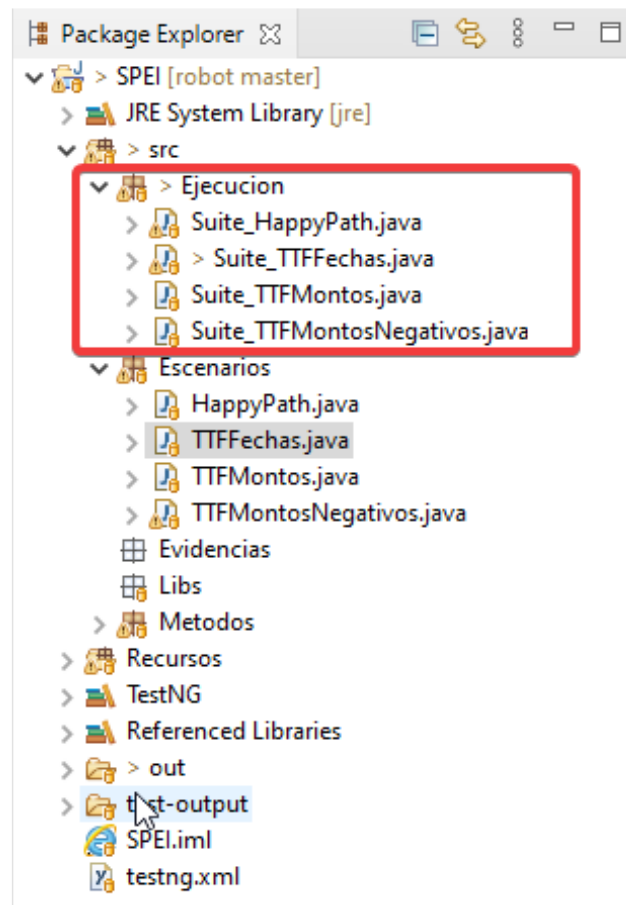


8. Hasta este momento ya se debe tener acceso a los scripts de pruebas automatizadas del proyecto clonado.



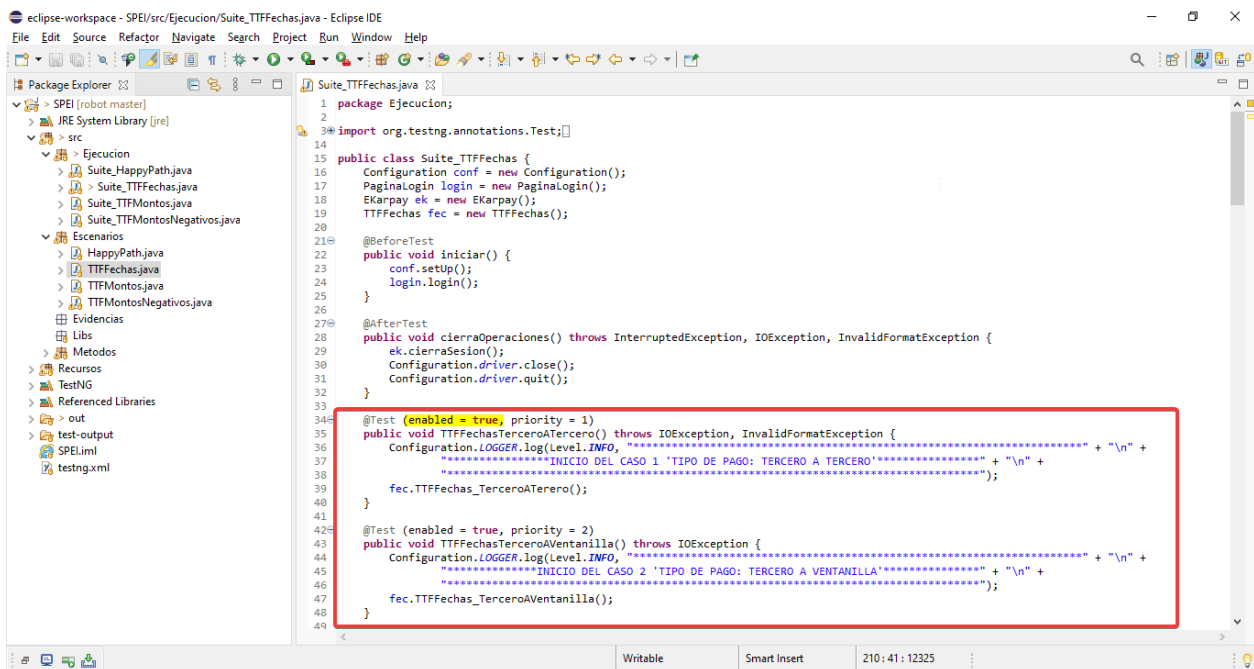
2 Ejecución de pruebas automatizadas.

1. Para ejecutar las pruebas automatizadas, tiene que ingresar a las clases que se encuentran dentro de la ruta donde se encuentran las validaciones disponibles para su ejecución.



2. Para mostrar el ejemplo dentro de este documento, se utilizará la validación Test to Fail para la validación de fechas.

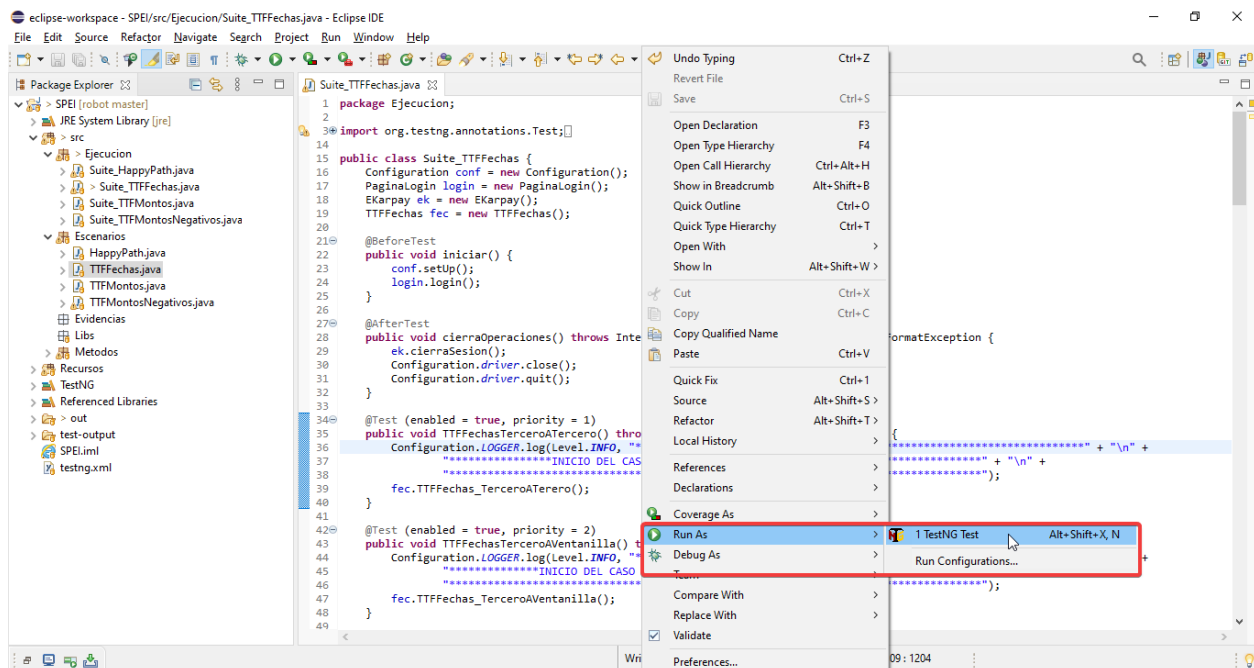
3. Dentro de cada clase de ejecución se muestran métodos que contienen la anotación “@Test”, dicha anotación indica que el método que se encuentra debajo se trata de un caso de prueba a ser ejecutado.
4. Cada anotación tienen sub-métodos llamados “enabled” y “priority”, si se quiere ejecutar un caso de prueba ya definido solo hay que cambiar el método ‘enabled’ y colocarlo como ‘true’, en caso de que no quiera ejecutar dicha validación hay que colocar un “false”, el método “priority” no se debe cambiar.



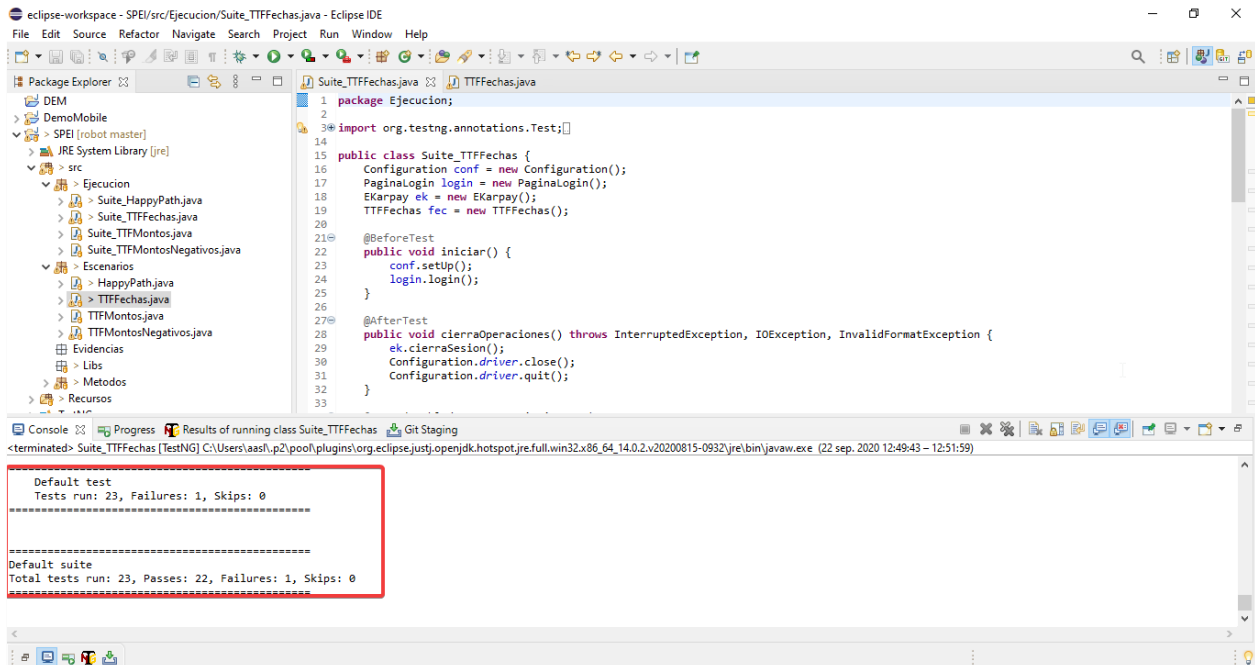
```
1 package Ejecucion;
2
3 import org.testng.annotations.Test;
4
5
6
7
8
9
10
11
12
13
14
15 public class Suite_TTFechas {
16     Configuration conf = new Configuration();
17     PaginaLogin login = new PaginaLogin();
18     EKarpay ek = new EKarpay();
19     TTFechas fec = new TTFechas();
20
21     @BeforeTest
22     public void iniciar() {
23         conf.setUp();
24         login.login();
25     }
26
27     @AfterTest
28     public void cierraOperaciones() throws InterruptedException, IOException, InvalidFormatException {
29         ek.cierraSesion();
30         Configuration.driver.close();
31         Configuration.driver.quit();
32     }
33
34     @Test(enabled = true, priority = 1)
35     public void TTFechasTerceroATercero() throws IOException, InvalidFormatException {
36         Configuration.LOGGER.log(Level.INFO, "*****INICIO DEL CASO 1 'TIPO DE PAGO: TERCERO A TERCERO'*****" + "\n" +
37             "*****");
38         fec.TTFechas_TerceroATercero();
39     }
40
41     @Test(enabled = true, priority = 2)
42     public void TTFechasTerceroAVentanilla() throws IOException {
43         Configuration.LOGGER.log(Level.INFO, "*****INICIO DEL CASO 2 'TIPO DE PAGO: TERCERO A VENTANILLA'*****" + "\n" +
44             "*****");
45         fec.TTFechas_TerceroAVentanilla();
46     }
47
48
49 }
```

5. Para ejecutar las validaciones que se hayan definido, haga clic derecho dentro de la clase de ejecución y seleccione la opción 'Run As', acto seguido elija la opción 'TestNG Test' y espere a que la ejecución se efectúe. Precondiciones para probar:

- VPN Activa.
- Ambientes estables.
- Usuario y Contraseña Parametrizados (Clase Login).



6. Una vez terminada la ejecución de pruebas, la consola muestra un recuento de casos de prueba que se ejecutaron, así como los casos de prueba que pasaron y fallaron.



```
 eclipse-workspace - SPE/src/Ejecucion/Suite_TTFFechas.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  DEM
  > DemoMobile
  > SPE [robot master]
  > JRE System Library [jre]
  > Ejecucion
    > Suite_HappyPath.java
    > Suite_TTFFechas.java
    > Suite_TTFMontos.java
    > Suite_TTFMontosNegativos.java
  > Escenarios
    > HappyPath.java
    > TTFFechas.java
    > TTFMontos.java
    > TTFMontosNegativos.java
  > Evidencias
  > Libs
  > Metodos
  > Recursos

Suite_TTFFechas.java
1 package Ejecucion;
2
3 import org.testng.annotations.Test;
4
14
15 public class Suite_TTFFechas {
16     Configuration conf = new Configuration();
17     PaginaLogin login = new PaginaLogin();
18     EKarpay ek = new EKarpay();
19     TTFFechas fec = new TTFFechas();
20
21 @BeforeTest
22 public void iniciar() {
23     conf.setUp();
24     login.login();
25 }
26
27 @AfterTest
28 public void cierraOperaciones() throws InterruptedException, IOException, InvalidFormatException {
29     ek.cierraSesion();
30     Configuration.driver.close();
31     Configuration.driver.quit();
32 }
33 }

Console
Progress Results of running class Suite_TTFFechas Git Staging
<terminated> Suite_TTFFechas [TestNG] C:\Users\laash\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.14.0.2.v20200815-0932\jre\bin\javaw.exe (22 sep. 2020 12:49:43 - 12:51:59)

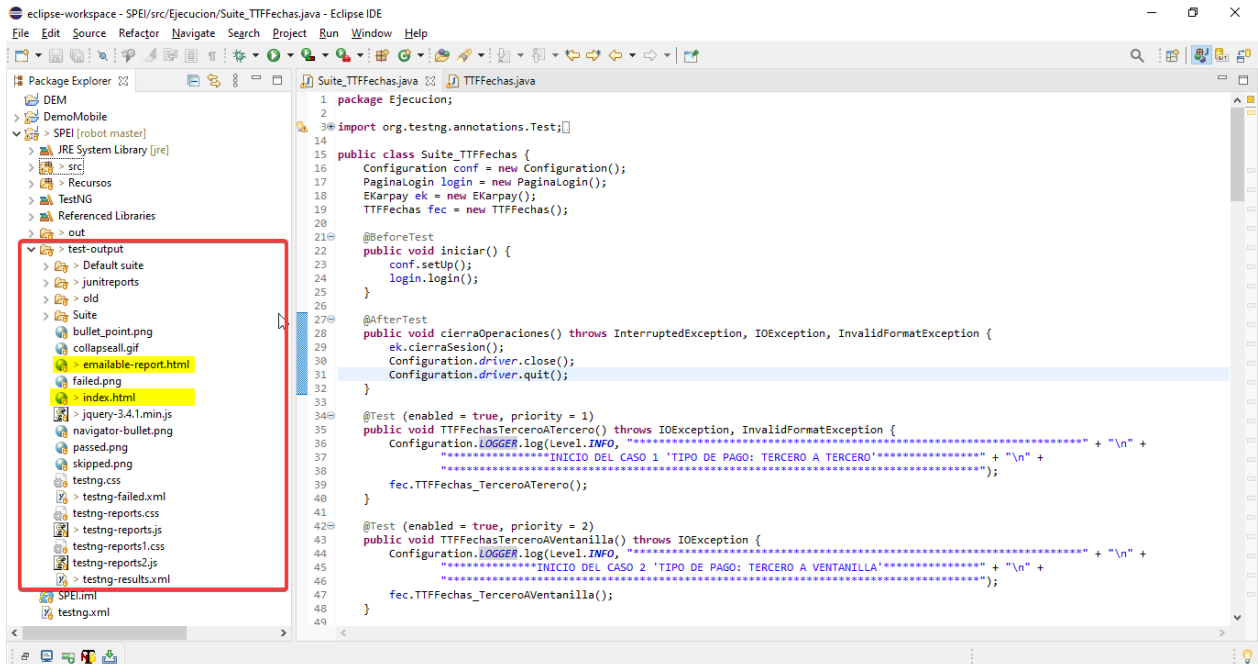
Default test
Tests run: 23, Failures: 1, Skips: 0
=====
Default suite
Total tests run: 23, Passes: 22, Failures: 1, Skips: 0
=====
```

2.1 Generación de reporte de pruebas

7. La estructura del proyecto genera una carpeta llamada 'test-output' donde dentro se pueden encontrar diferentes vistas sobre los reportes que la herramienta genera:

Las más importantes y que muestran mayor información son:

- `emailable-report.html`
- `Index.html`



8. emailable-report.html, muestra un recuento de todos aquellos casos de prueba que fallaron y los muestra en color rojo, en verde, se muestran los casos de prueba que pasaron.
9. En la parte superior se muestra el tiempo total que se tomó para ejecutar la suite de casos de prueba completos.

File Edit Navigate Search Project Run Window Help

Package Explorer Suite_TTFechas.java TTFechas.java TestNG Report

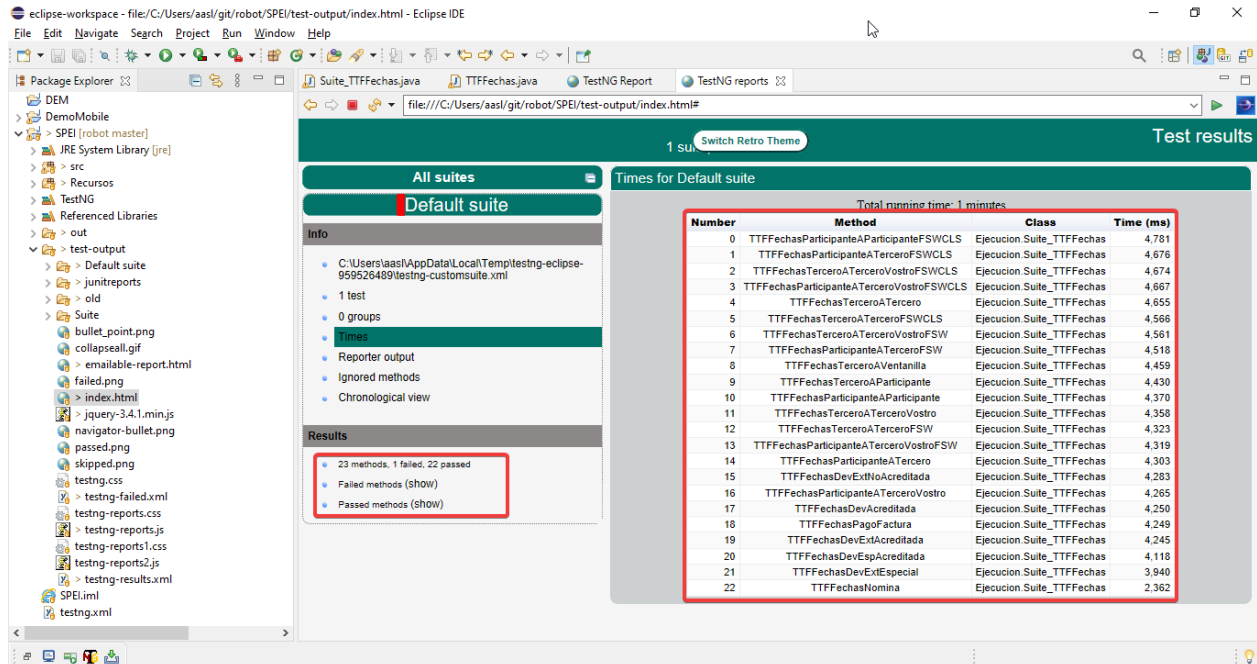
file:///C:/Users/aasil/git/robot/SPEI/test-output/emailable-report.html

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	22	0	0	1	131,741		

Class	Method	Start	Time (ms)
Default suite			
Default test — failed			
ExecutionSuite_TTFechas	TTFechasNaming	1600797061208	2362
Default test — passed			
ExecutionSuite_TTFechas	TTFechasBulletPointsAcordado	1600797072111	4250
	TTFechasBulletPointsAcordado	1600797080609	4118
	TTFechasBulletPointsAcordado	1600797076342	4245
	TTFechasBulletPointsAcordado	1600797064728	3940
	TTFechasBulletPointsAcordado	1600797067826	4283
	TTFechasBulletPointsAcordado	1600797063574	4249
	TTFechasBulletPointsAcordado	1600797039109	4370
	TTFechasBulletPointsAcordado	1600797107256	4781
	TTFechasBulletPointsAcordado	1600797030537	4303
	TTFechasBulletPointsAcordado	1600797023568	4518
	TTFechasBulletPointsAcordado	1600797097911	4676
	TTFechasBulletPointsAcordado	1600797034842	4245
	TTFechasBulletPointsAcordado	1600797056887	4319
	TTFechasBulletPointsAcordado	1600797102358	4667
	TTFechasBulletPointsAcordado	1600797026104	4430
	TTFechasBulletPointsAcordado	1600797012623	4655
	TTFechasBulletPointsAcordado	1600797045481	4323
	TTFechasBulletPointsAcordado	1600797088669	4366
	TTFechasBulletPointsAcordado	1600797021744	4358
	TTFechasBulletPointsAcordado	1600797047805	4561
	TTFechasBulletPointsAcordado	1600797093226	4674

Finalizado

10.index.html, muestra los casos de prueba individuales y el tiempo de ejecución por cada uno de ellos. Así como una vista donde podremos ver los casos de prueba pasados y los casos de prueba fallidos.



The screenshot shows the Eclipse IDE interface with the TestNG Report open. The Package Explorer on the left shows the project structure, including the 'test-output' directory. The main window displays the 'Test results' for the 'Default suite'. A table lists 23 methods, with 22 passed and 1 failed. The failed method is 'TTFFechasDevExtNoAcreditada'.

Number	Method	Class	Time (ms)
0	TTFFechasParticipanteAParticipanteFSWCLS	Ejecucion_Suite_TTFFechas	4,781
1	TTFFechasParticipanteATerceroFSWCLS	Ejecucion_Suite_TTFFechas	4,676
2	TTFFechasTerceroATerceroVostroFSWCLS	Ejecucion_Suite_TTFFechas	4,674
3	TTFFechasParticipanteATerceroVostroFSWCLS	Ejecucion_Suite_TTFFechas	4,667
4	TTFFechasTerceroATercero	Ejecucion_Suite_TTFFechas	4,655
5	TTFFechasTerceroATerceroFSWCLS	Ejecucion_Suite_TTFFechas	4,566
6	TTFFechasTerceroATerceroVostroFSW	Ejecucion_Suite_TTFFechas	4,561
7	TTFFechasParticipanteATerceroFSW	Ejecucion_Suite_TTFFechas	4,518
8	TTFFechasTerceroAVentanilla	Ejecucion_Suite_TTFFechas	4,459
9	TTFFechasTerceroAParticipante	Ejecucion_Suite_TTFFechas	4,430
10	TTFFechasParticipanteAParticipante	Ejecucion_Suite_TTFFechas	4,370
11	TTFFechasTerceroATerceroVostro	Ejecucion_Suite_TTFFechas	4,356
12	TTFFechasTerceroATerceroFSW	Ejecucion_Suite_TTFFechas	4,323
13	TTFFechasParticipanteATerceroVostroFSW	Ejecucion_Suite_TTFFechas	4,319
14	TTFFechasParticipanteATercero	Ejecucion_Suite_TTFFechas	4,303
15	TTFFechasDevExtNoAcreditada	Ejecucion_Suite_TTFFechas	4,283
16	TTFFechasParticipanteATerceroVostro	Ejecucion_Suite_TTFFechas	4,265
17	TTFFechasDevAcreditada	Ejecucion_Suite_TTFFechas	4,250
18	TTFFechasPagoFactura	Ejecucion_Suite_TTFFechas	4,249
19	TTFFechasDevExtAcreditada	Ejecucion_Suite_TTFFechas	4,245
20	TTFFechasDevEspAcreditada	Ejecucion_Suite_TTFFechas	4,118
21	TTFFechasDevExtEspecial	Ejecucion_Suite_TTFFechas	3,940
22	TTFFechasNomina	Ejecucion_Suite_TTFFechas	2,362

2.2 Captura de Evidencias

11. Se debe contar con una ruta dentro del proyecto (Carpeta llamada “Evidencias”) para guardar capturas de pantalla de la ejecución realizada sobre las validaciones realizadas.

NOTA: se está trabajando para poder armar el archivo de evidencia de manera automática (Word, PDF, Excel, etc).

