

Objetivo

Presentar desde el ámbito tecnológico la instalación y configuración del software necesario para la construcción y ejecución del robot de automatización, siendo referente el aplicativo orquídea del Banco Bilbao Vizcaya Argentaria BBVA.

Herramientas

- Eclipse Oxygen Version: Oxygen.1a Release (4.7.1a), Build id: 20171005-1200
- JAVA version 8 Update 151 64 o 32 bits.
- Maven version 1.8
- Serenity version 1.6.3
- Junit version 4.12
- Serenity – Cucumber version 1.5.9

PASO 1

Verificar que en la carpeta C:\Users\usuariodeequipos\m2 esté vacía si existe la carpeta repository se procede a borrarla. Cerrando Eclipse previamente.

CONSTRUCCIÓN DEL FRAMEWORK EN ECLIPSE

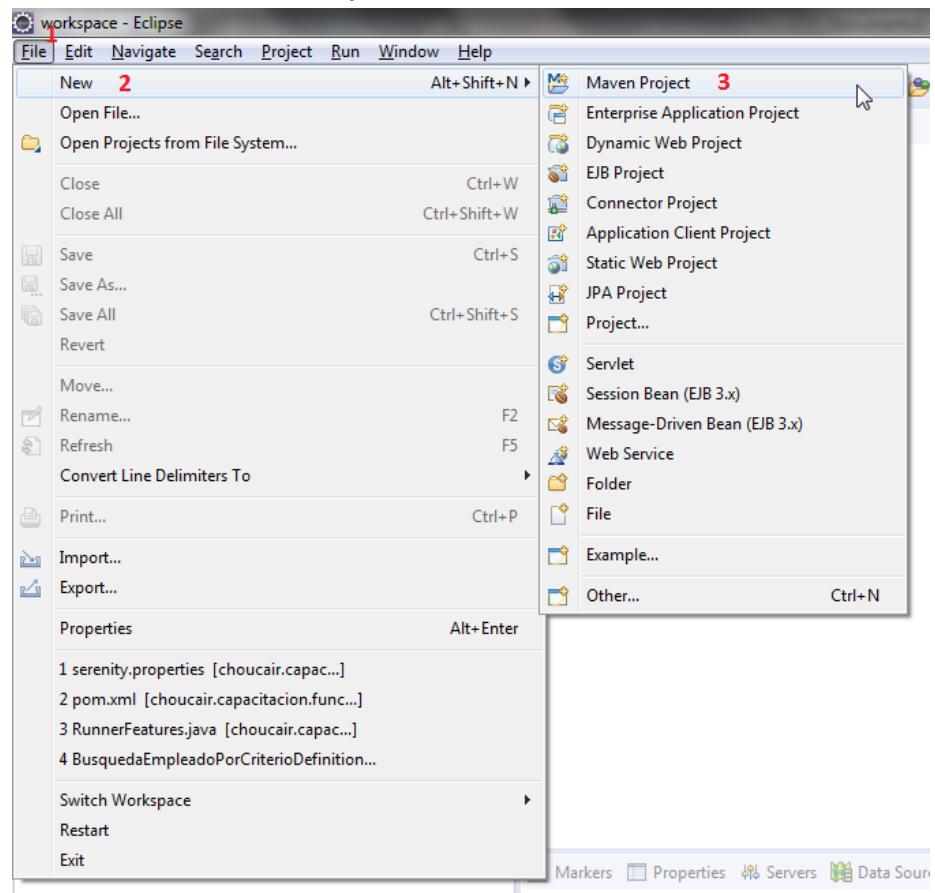
Creación de un Proyecto

ESTRUCTURA ESTANDAR

Proyecto Línea	com.choucair.formacion						
	Aplicativo 1	Planet					
		Modulo 1	Búsqueda				
			.feature	BúsquedaEmpleado			
				scenario = caso de prueba	Buscar Analistas de Prueba	Tipo CP	SmokeTest
				scenario = caso de prueba	Buscar Automatizadores	Tipo CP	RutaCrítica
			+ .feature	BúsquedaEmpresas			
				scenario = caso de prueba	Buscar empresas medellín	Tipo CP	SmokeTest
		Modulo 2	Plantilla Documentos				
			.feature	DescargaPlantillas			
				scenario = caso de prueba	Decargar Plantilla de Presentaciones	Tipo CP	SmokeTest
	Aplicativo 2	KM					
		Modulo 1	Consultas				
			.feature	Consulta Cursos			
				scenario = caso de prueba	Consulta cursos Analista x	Tipo CP	RutaCrítica
				scenario = caso de prueba	Consulta Cursos Administrativos	Tipo CP	Estandar
Proyecto Cliente	com.choucair.unico						
	Aplicativo 1	Venta en Línea					
		Modulo 1	Compras				
			.feature	Ver mi orden			
				scenario = caso de prueba	Aplicar cupón de descuento	Tipo CP	SmokeTest
				scenario = caso de prueba	Aumentar Cantidad	Tipo CP	RutaCrítica
				scenario = caso de prueba	Disminuir Cantidad	Tipo CP	SmokeTest
				scenario = caso de prueba	Bajar el carro	Tipo CP	RutaCrítica
			+ .feature	Logística			
				scenario = caso de prueba	Precio Fletes	Tipo CP	
				scenario = caso de prueba	Días	Tipo CP	
			.feature	Pagar			
				scenario = caso de prueba	Pago con TC	Tipo CP	
				scenario = caso de prueba	Pago con TD	Tipo CP	

Puede hacerse de dos maneras:

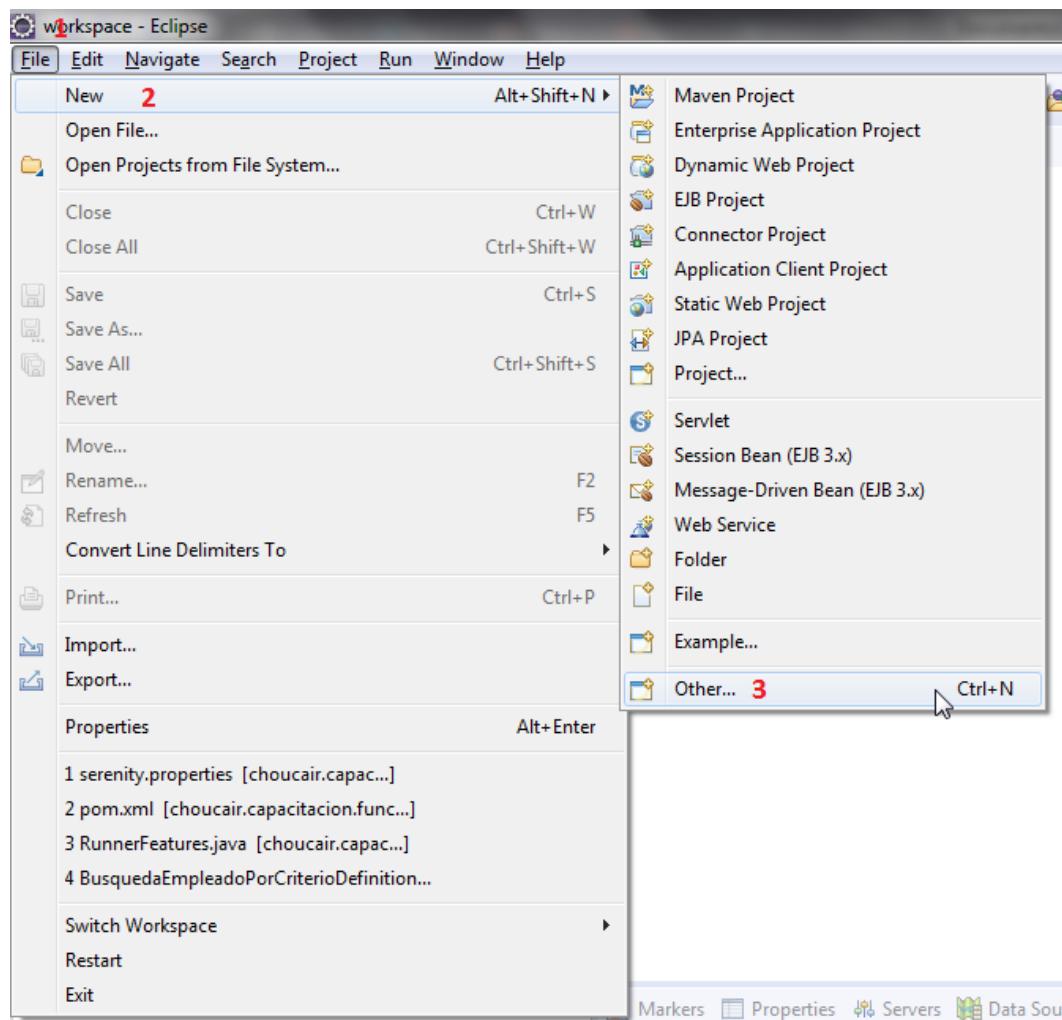
1. Seleccionando directamente el **Project Maven**



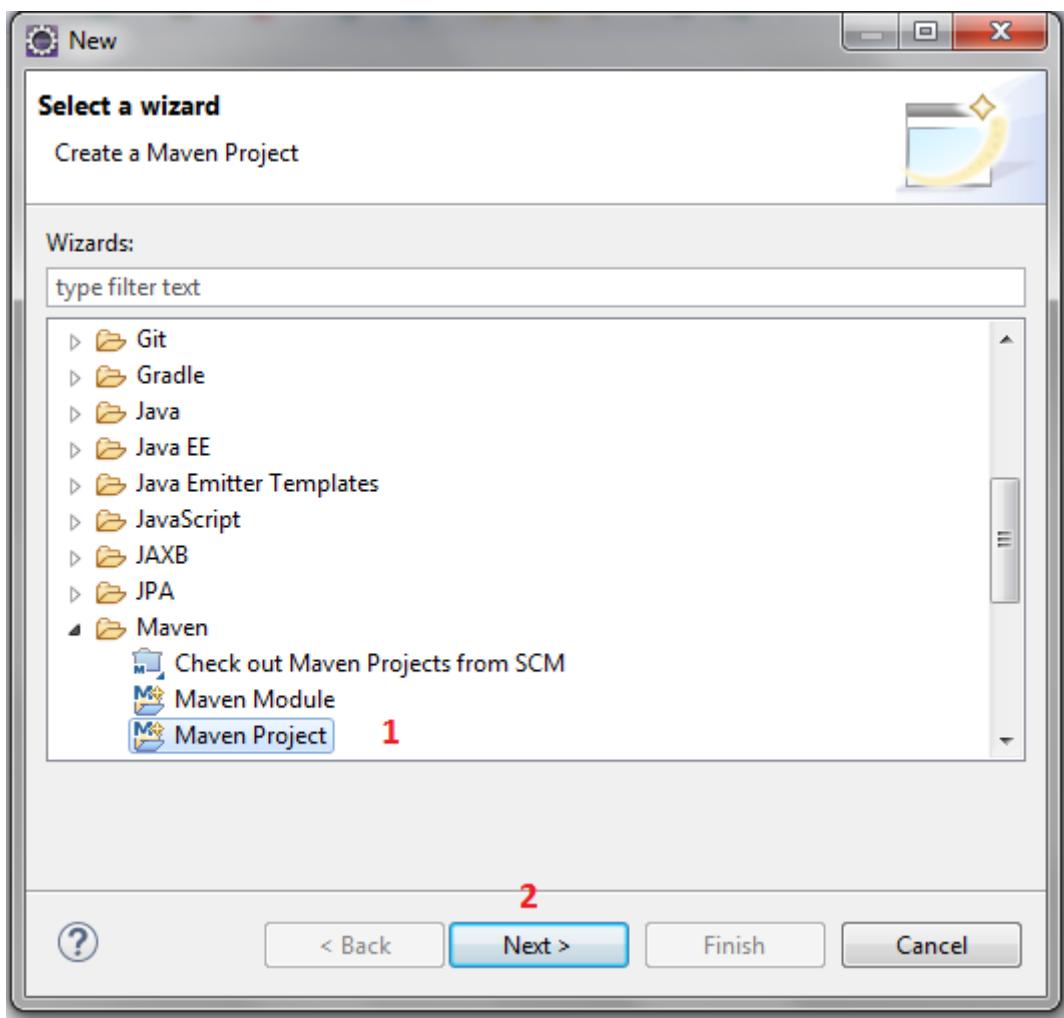
2. Seleccionando Other...

Tiene dos pasos

- Primer Paso

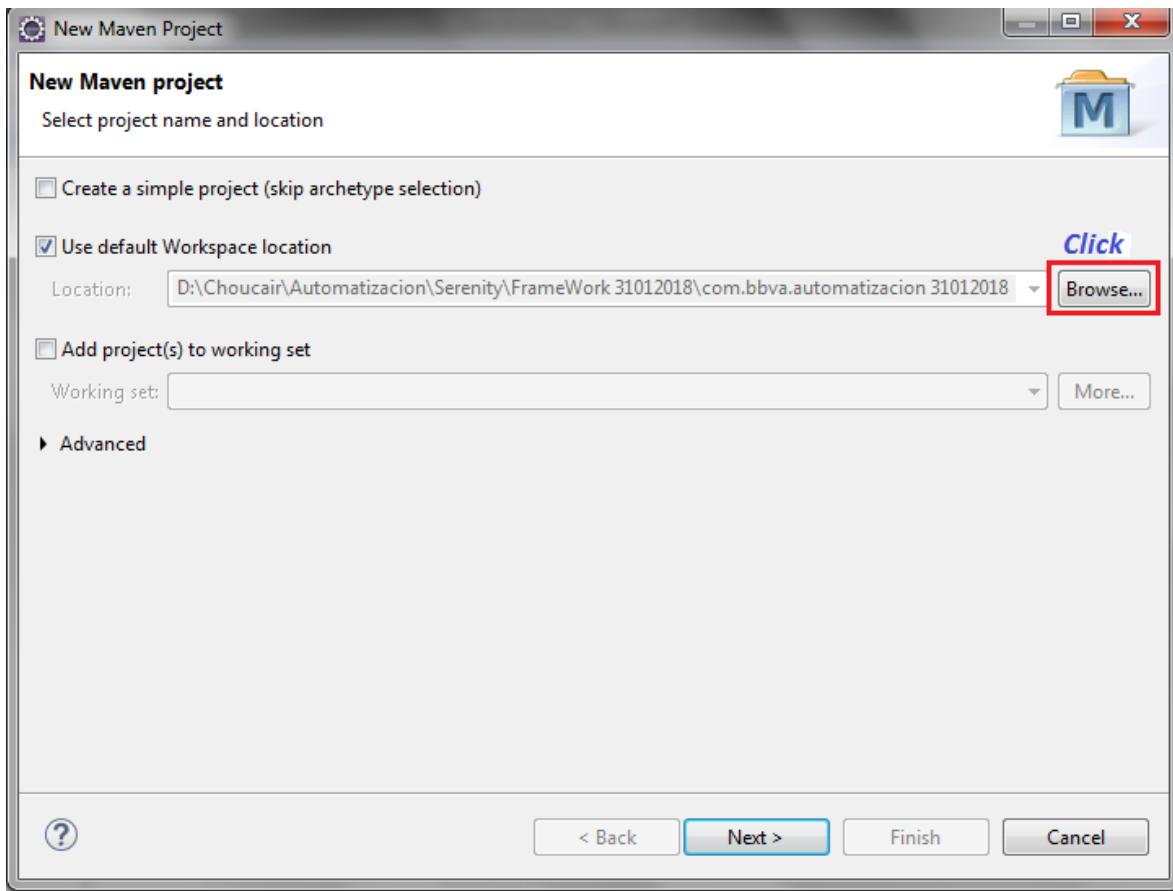


- Segundo paso

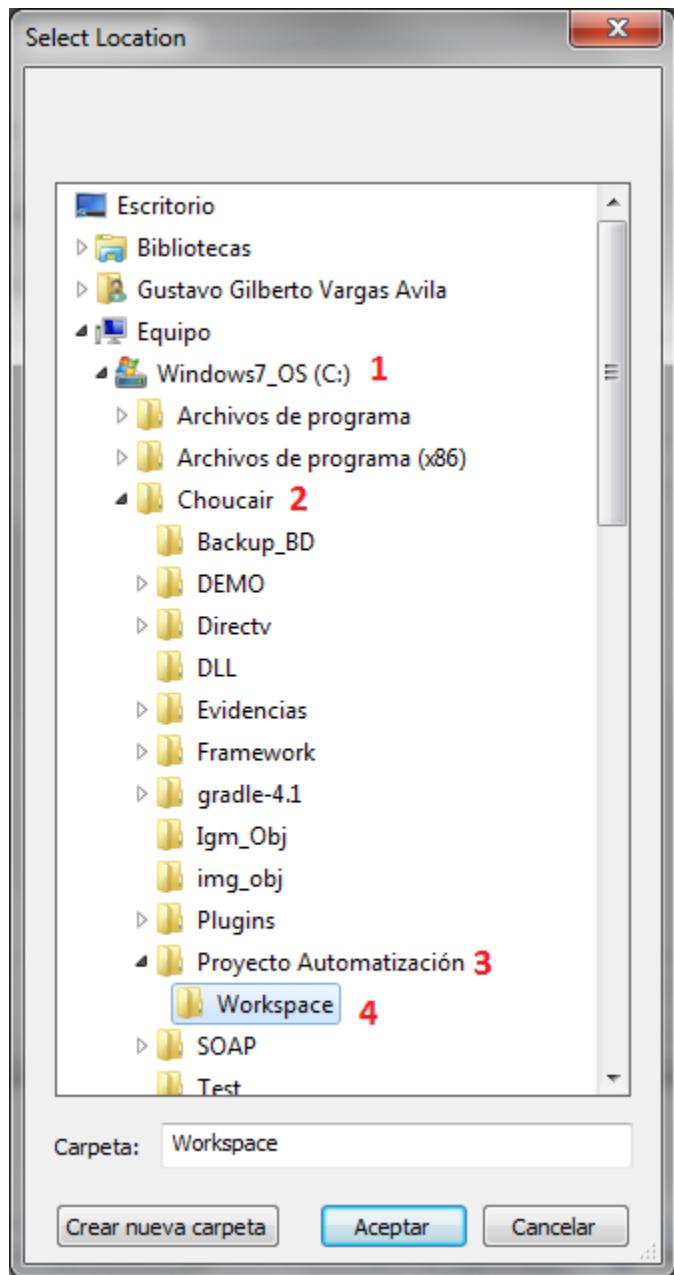


Las dos maneras llevan a la siguiente pantalla:

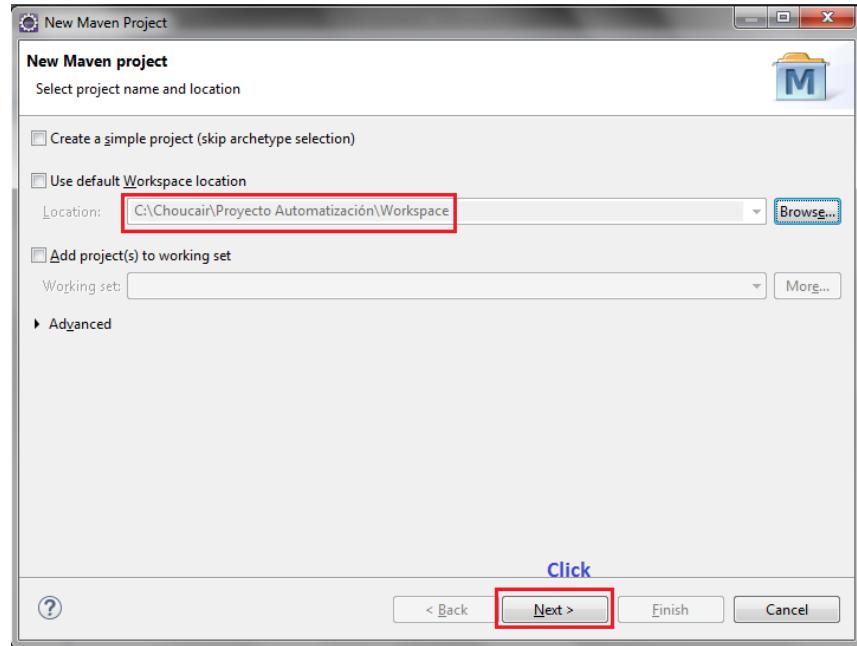
Aquí se selecciona el Workspace o carpeta que contendrá el proyecto, se puede dejar en la carpeta que muestra por defecto, pero se recomienda construirla en la siguiente ruta:
C:\Choucair\Proyecto Automatización\Workspace



Se selecciona la ubicación del nuevo Workspace, deben crear las carpetas que menciona la ruta si no están creadas

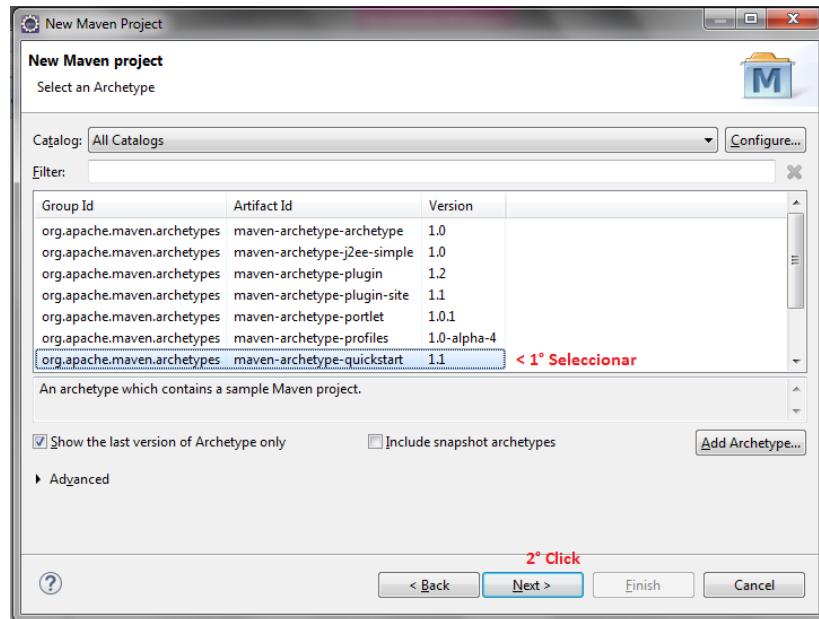


La pantalla se debe visualizar, donde se muestra el Path del workspace como indica la siguiente imagen, donde se da click para continuar:



SELECCIONANDO MAVEN-ARCHETYPE-QUICKSTART

Se visualizará la siguiente pantalla: Seleccionar la opción “**maven-archetype-quickstart 1.1**” debe estar seleccionada por defecto de esta forma se crea un proyecto Maven simple:



Se crea la estructura del proyecto: Cliente + . + Línea o área + . + tipo automatización:

Choucair.capacitacion.funcional

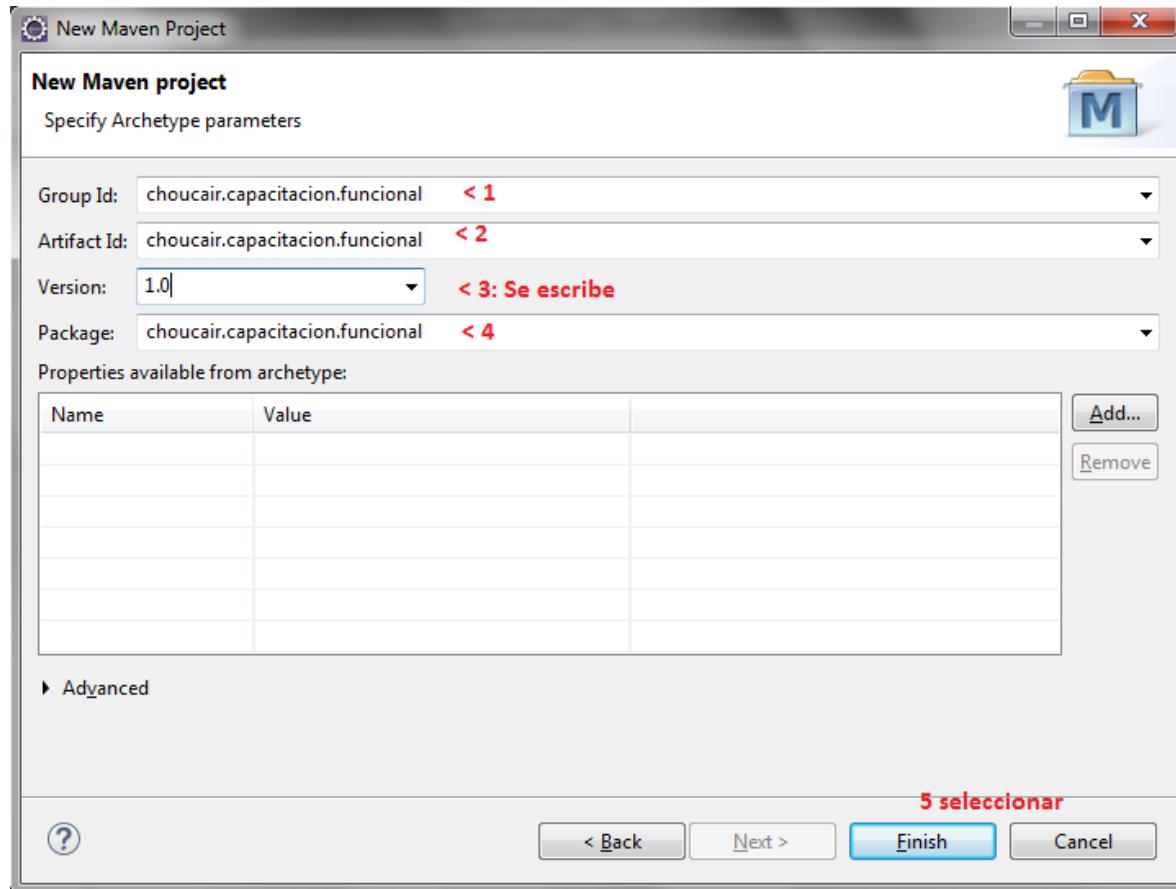
Ejemplos:

1° claro.fija.funcional ---- 2° claro.fija.servicios ---- 3° bbva.canales.funcional

Deben tener el mismo nombre o contenido los siguientes campos:

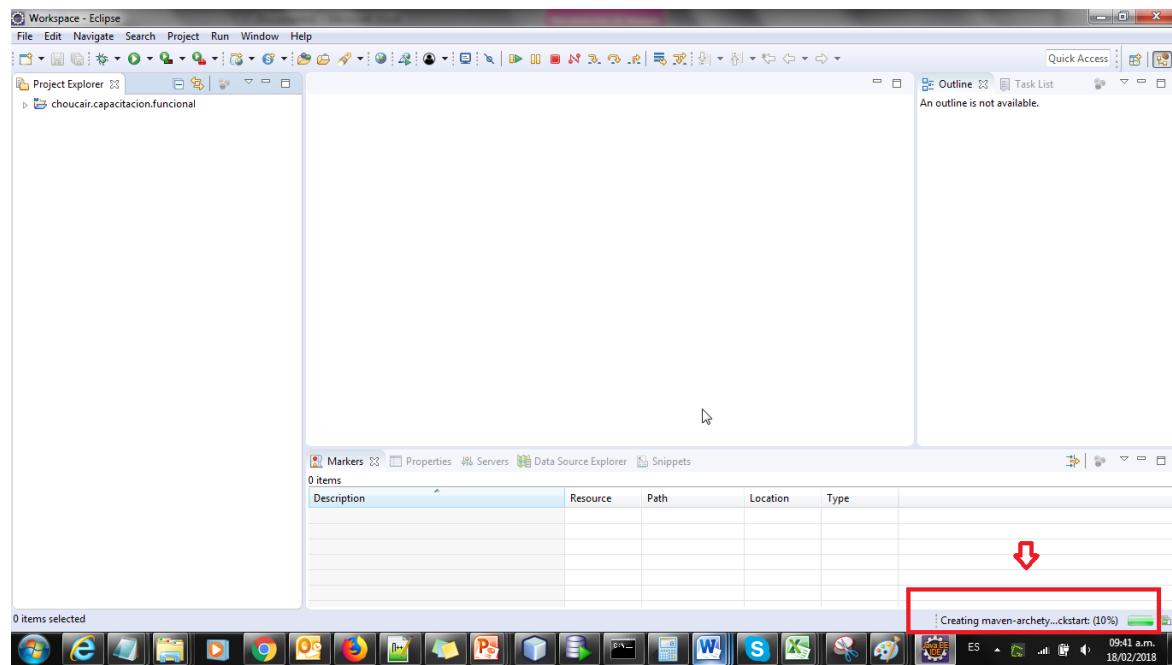
- *Grupo Id:*
- *Artifact Id:*
- *Package:*

En el campo Version se escribe **1.0** como indica la imagen

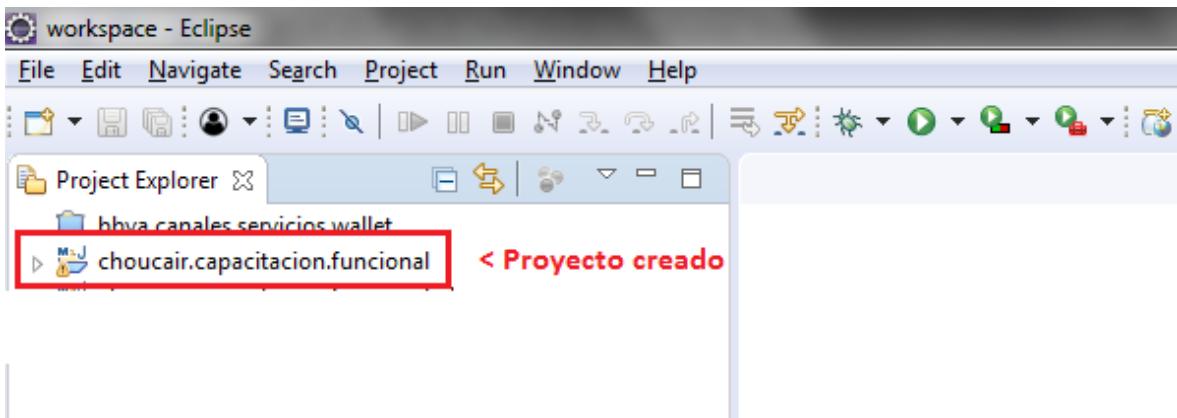


Se da click en Finish

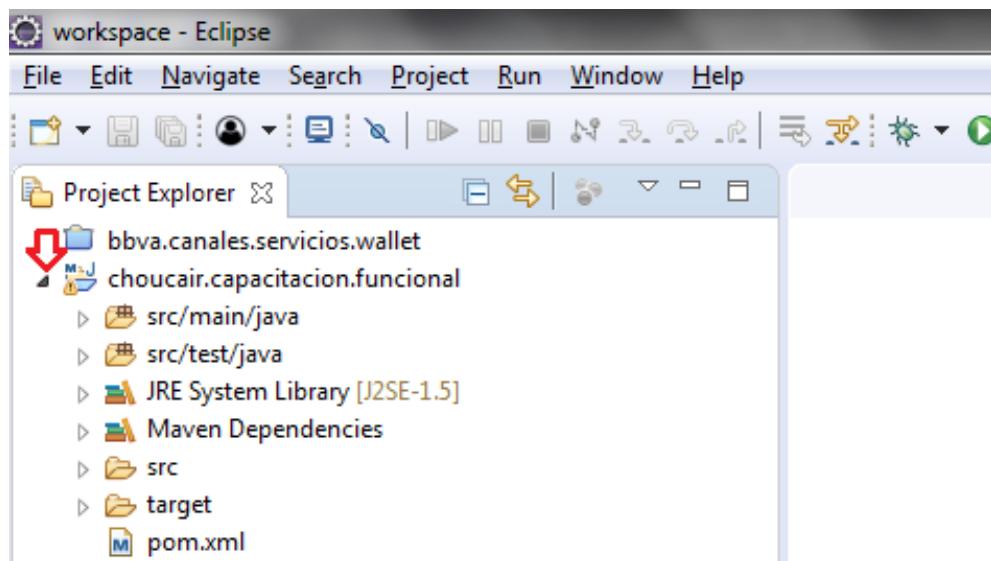
Eclipse mostrará el avance de la creación del proyecto



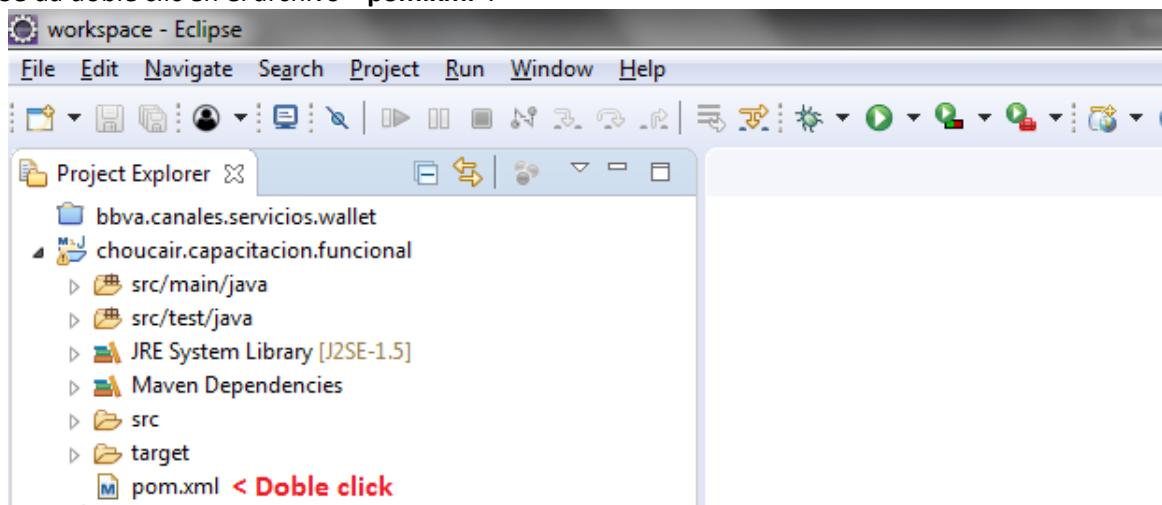
Después de unos instantes eclipse mostrará el proyecto creado como indica la imagen:



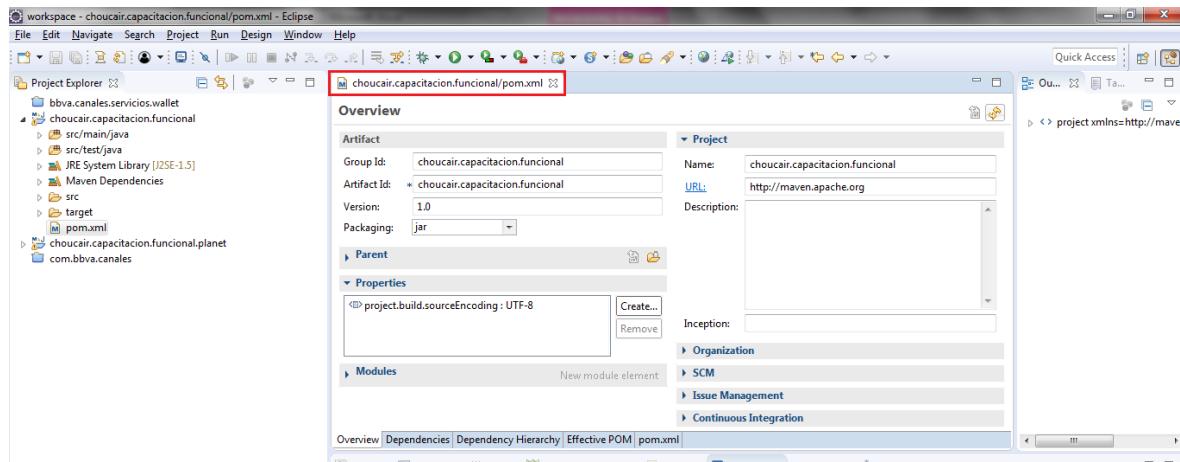
Se despliega el proyecto dando click en la flecha como indica la imagen:



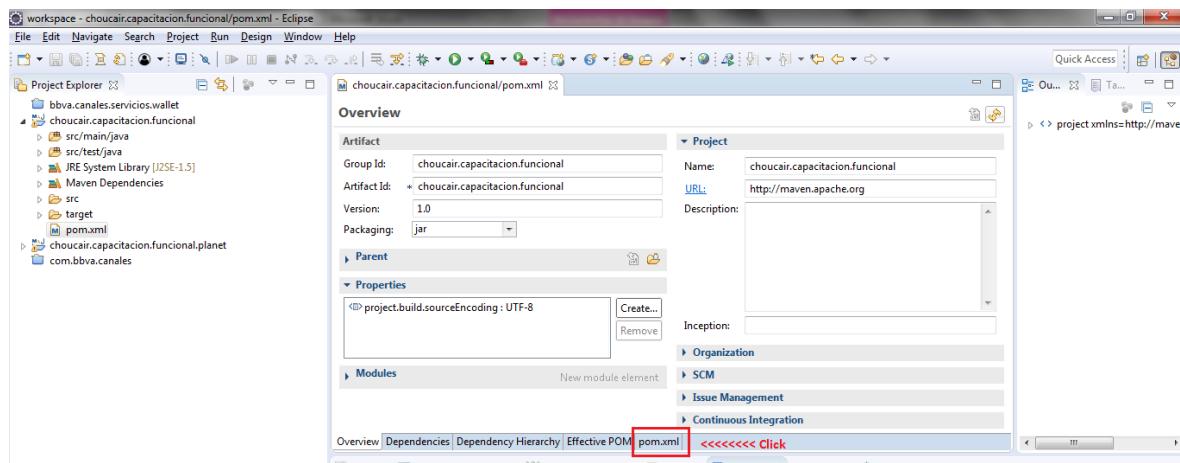
Se da doble clic en el archivo “**pom.xml**”:



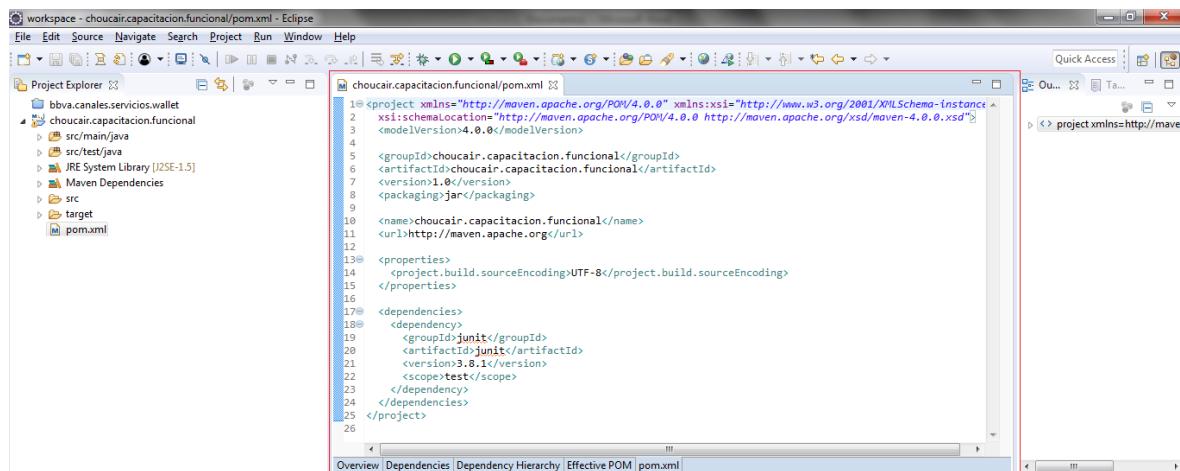
En la pantalla se visualizará como indica la imagen:



Lo que se hará a continuación es determinar las diferentes propiedades y dependencias que se requiere para elaborar, gestionar y ejecutar el framework, para visualizar aquellas que se encuentran por defecto se selecciona la etiqueta inferior de nombre “**pom.xml**”:



La pantalla se visualizará como indica la imagen:



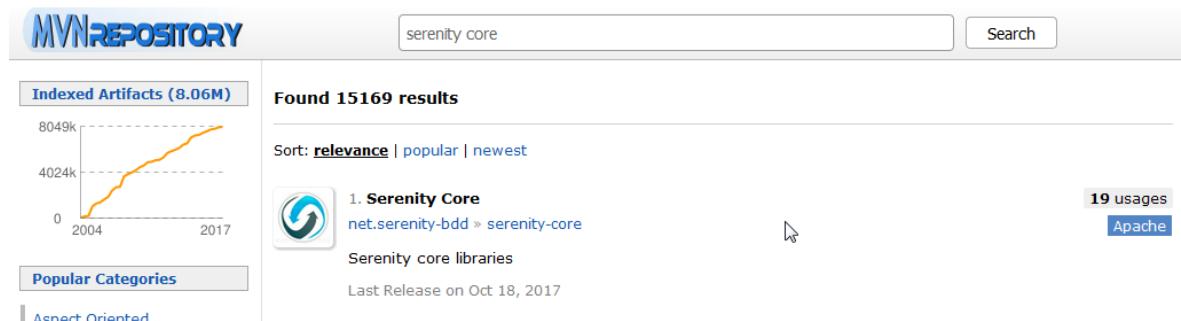
Se procede a modificar esta sección con el fin de descargar los archivos que se utilizarán en el proyecto, existen varias formas de hacerlo, bien se puede construir desde cero las dependencias ingresando las etiquetas pertinentes siempre y cuando concuerden con el repositorio de maven, copiar un archivo **pom.xml** anteriormente constituido en otro proyecto o copiar uno existente.

INFORMACIÓN IMPORTANTE

Es fundamental que en cada proyecto se instauren repositorios actualizados o en su defecto estables para cuyo caso se puede apreciar el número de usuarios que actualmente lo usan.

En la imagen se muestra un ejemplo de cómo añadir el core de Serenity:

Se accede desde la URL <https://mvnrepository.com/artifact/net.serenity-bdd>



<https://mvnrepository.com/artifact/net.serenity-bdd/serenity-core>

Se sugiere buscar la versión que más usuarios tiene

	Version	Repository	Usages	Date
1.6.x	1.6.7-rc.1	Central	<div style="width: 10px;"></div> 10	(Oct, 2017)
	1.6.6	Central	<div style="width: 10px;"></div> 10	(Oct, 2017)
	1.6.6-rc.2	Central	<div style="width: 10px;"></div> 10	(Oct, 2017)
	1.6.5	Central	<div style="width: 11px;"></div> 11	(Oct, 2017)
	1.6.4	Central	<div style="width: 10px;"></div> 10	(Oct, 2017)
	1.6.4-rc.2	Central	<div style="width: 9px;"></div> 9	(Oct, 2017)
	1.6.3 Click	Central	<div style="width: 12px;"></div> 12	(Oct, 2017)
	1.6.1	Central	<div style="width: 11px;"></div> 11	(Oct, 2017)
	1.6.0	Central	<div style="width: 12px;"></div> 12	(Sep, 2017)
1.5.x	1.5.11	Central	<div style="width: 11px;"></div> 11	(Sep, 2017)
	1.5.10	Central	<div style="width: 10px;"></div> 10	(Sep, 2017)
	1.5.9	Central	<div style="width: 11px;"></div> 11	(Aug, 2017)
	1.5.8	Central	<div style="width: 10px;"></div> 10	(Aug, 2017)

Existen más usuario usando la versión 1.6.3 (12 usuarios) aunque ese mismo número de usuarios usa la versión 1.6.0 se tiende a usar la más actualizada dentro éstos dos es decir la 1.6.3.



Serenity Core > 1.6.3

Serenity core libraries

License	Apache 2.0
HomePage	https://github.com/serenity-bdd/serenity-core
Date	(Oct 02, 2017)
Files	Download (JAR) (3.4 MB)
Repositories	Central Sonatype Releases
Used By	19 artifacts

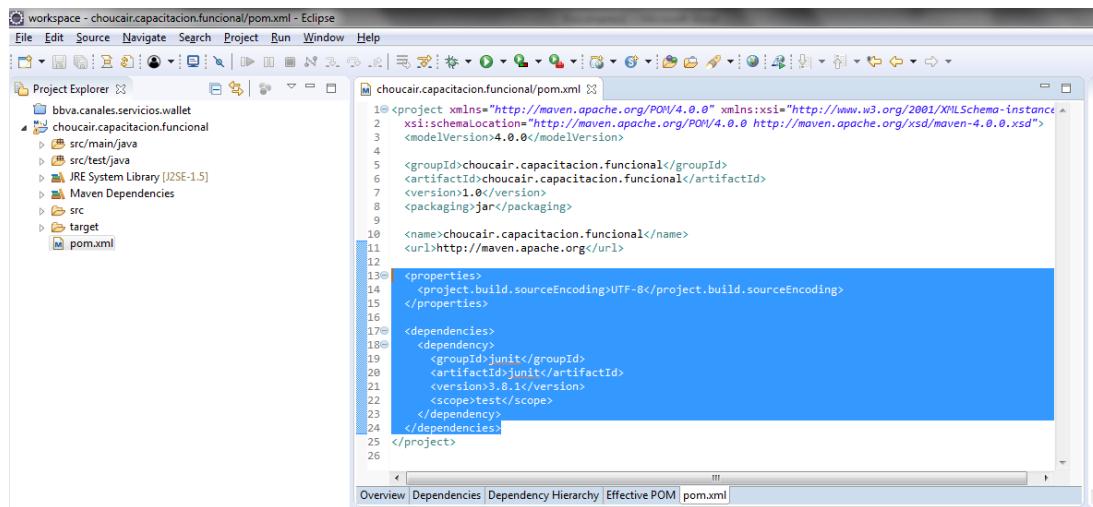
[Maven](#) [Gradle](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#) [Buildr](#)

```
<!-- https://mvnrepository.com/artifact/net.serenity-bdd/serenity-core -->
<dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-core</artifactId>
    <version>1.6.3</version>
</dependency>
```

Copiar y pegar en el **pom.xml** la dependencia que se muestra en el área de texto.

SE CONTINÚA CON EL EJERCICIO:

Las siguientes líneas (**naranja**) son copiadas y pegadas en el archivo pom.xml, reemplazando el contenido del archivo **pom.xml**, desde la etiqueta **<properties>** hasta antes de la etiqueta **</project>** como se muestra en la imagen.



```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <serenity.version>1.6.3</serenity.version> <!-- https://bintray.com/serenity/maven/serenity-core/ -->
        <serenity.maven.version>1.6.3</serenity.maven.version>
        <serenity.cucumber.version>1.5.9</serenity.cucumber.version> <!-- https://bintray.com/serenity/maven/serenity-cucumber/ -->
        <maven.compiler.source>1.8</maven.compiler.source>
            <maven.compiler.target>1.8</maven.compiler.target>
            <junit.version>4.12</junit.version>
    </properties>

    <!-- Define the Bintray repos for convenience -->
    <repositories>
        <repository>
            <id>serenity</id>
            <name>bintray</name>
            <url>http://dl.bintray.com/serenity/maven</url>
        </repository>
    </repositories>
    <pluginRepositories>
        <pluginRepository>
            <id>serenity</id>
            <name>bintray-plugins</name>
            <url>http://dl.bintray.com/serenity/maven</url>
        </pluginRepository>
    </pluginRepositories>

    <dependencies>
        <!-- https://mvnrepository.com/artifact/org.hamcrest/hamcrest-junit -->
        <dependency>
            <groupId>org.hamcrest</groupId>
            <artifactId>hamcrest-junit</artifactId>
```

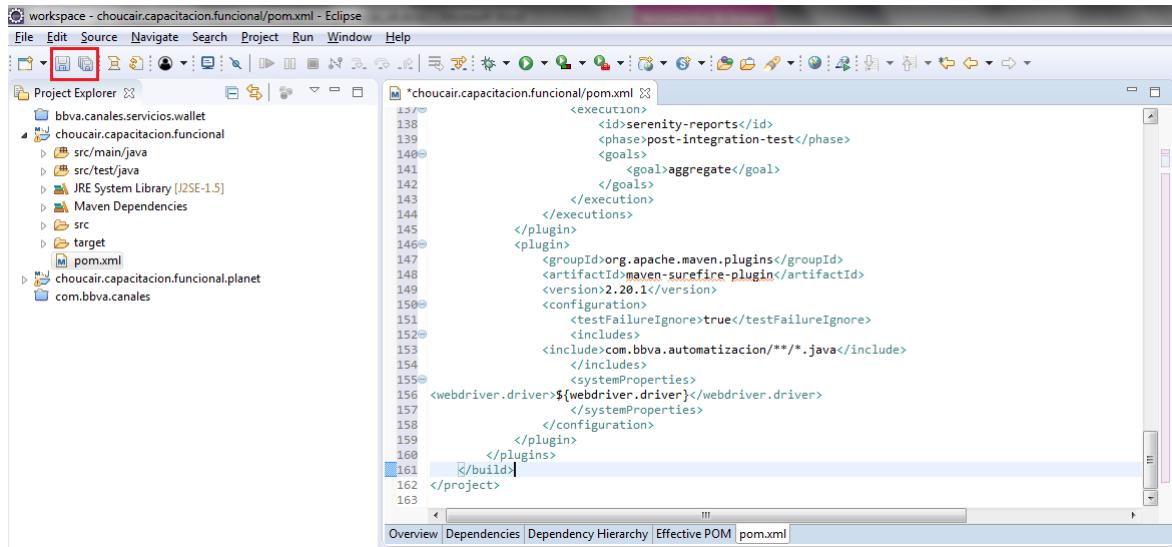
```
        <version>2.0.0.0</version>
        <scope>test</scope>
    </dependency>
<dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-core</artifactId>
    <version>${serenity.version}</version>
</dependency>
<dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-junit</artifactId>
    <version>${serenity.version}</version>
</dependency>
<dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-rest-assured</artifactId>
    <version>${serenity.version}</version>
</dependency>
<dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-cucumber</artifactId>
    <version>${serenity.cucumber.version}</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.6.1</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.codehaus.groovy</groupId>
    <artifactId>groovy-all</artifactId>
    <version>1.8.6</version>
</dependency>
<dependency>
    <groupId>org.assertj</groupId>
    <artifactId>assertj-core</artifactId>
    <version>3.8.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
        <artifactId>poi</artifactId>
        <version>3.17</version>
    </dependency>
    <dependency>
        <groupId>org.apache.poi</groupId>
        <artifactId>poi-ooxml</artifactId>
        <version>3.17</version>
    </dependency>
</dependency>
</dependencies>
<build>
    <plugins>
```

```

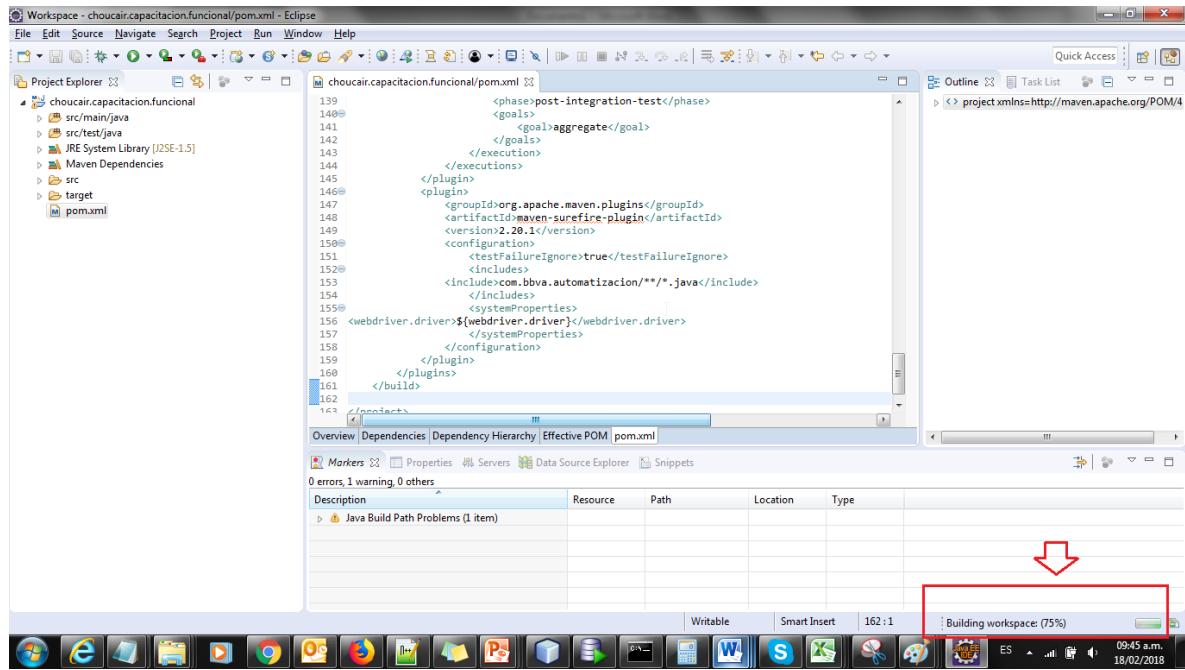
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>3.7.0</version>
    <executions>
        <execution>
            <id>default-compile</id>
            <configuration>
                <showDeprecation>true</showDeprecation>
                <showWarnings>true</showWarnings>
                <compilerArguments>
                    <source>${maven.compiler.source}</source>
                    <target>${maven.compiler.target}</target>
                </compilerArguments>
            </configuration>
        </execution>
        <execution>
            <id>default-testCompile</id>
            <configuration>
                <showDeprecation>true</showDeprecation>
                <showWarnings>true</showWarnings>
                <compilerArguments>
                    <source>${maven.compiler.source}</source>
                    <target>${maven.compiler.target}</target>
                </compilerArguments>
            </configuration>
        </execution>
    </executions>
</plugin>
<plugin>
    <groupId>net.serenity-bdd.maven.plugins</groupId>
    <artifactId>serenity-maven-plugin</artifactId>
    <version>1.5.9</version>
    <executions>
        <execution>
            <id>serenity-reports</id>
            <phase>post-integration-test</phase>
            <goals>
                <goal>aggregate</goal>
            </goals>
        </execution>
    </executions>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-surefire-plugin</artifactId>
    <version>2.20.1</version>
    <configuration>
        <testFailureIgnore>true</testFailureIgnore>
        <includes>
            <include>com.bbva.automatizacion/**/*.{java}</include>
        </includes>
        <systemProperties>
            <webdriver.driver>${webdriver.driver}</webdriver.driver>
        </systemProperties>
    </configuration>
</plugin>
</plugins>
</build>

```

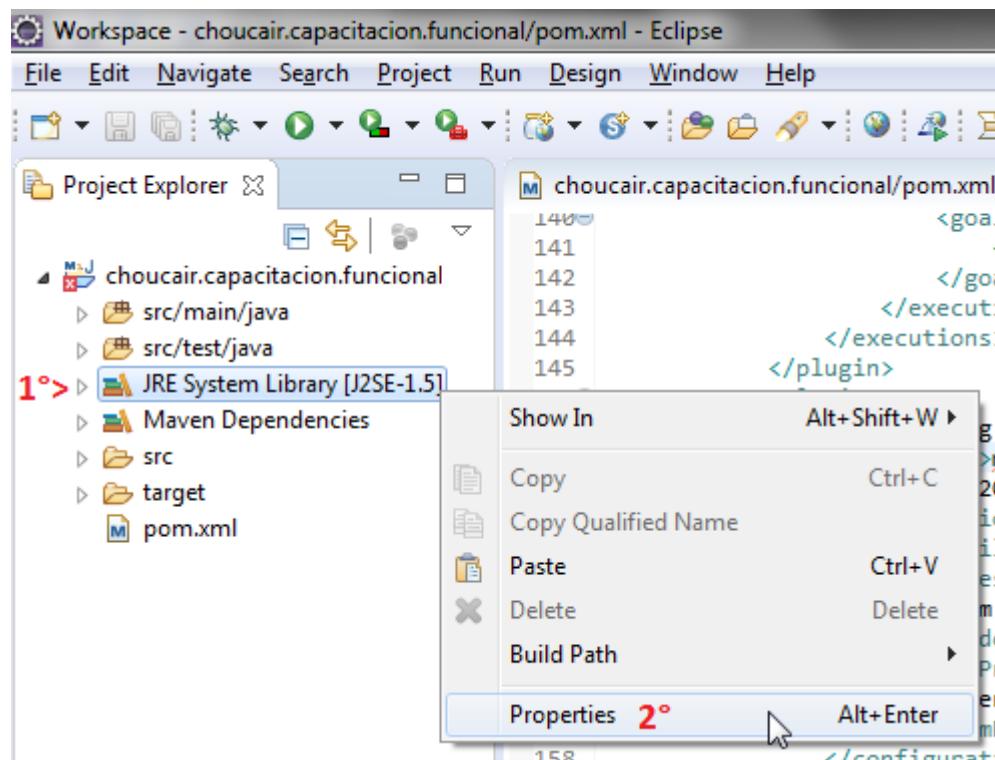
Guardar



Se debe esperar un tiempo mientras hace las descargas de las dependencias, habrá una barra indicando el proceso como indica la imagen:

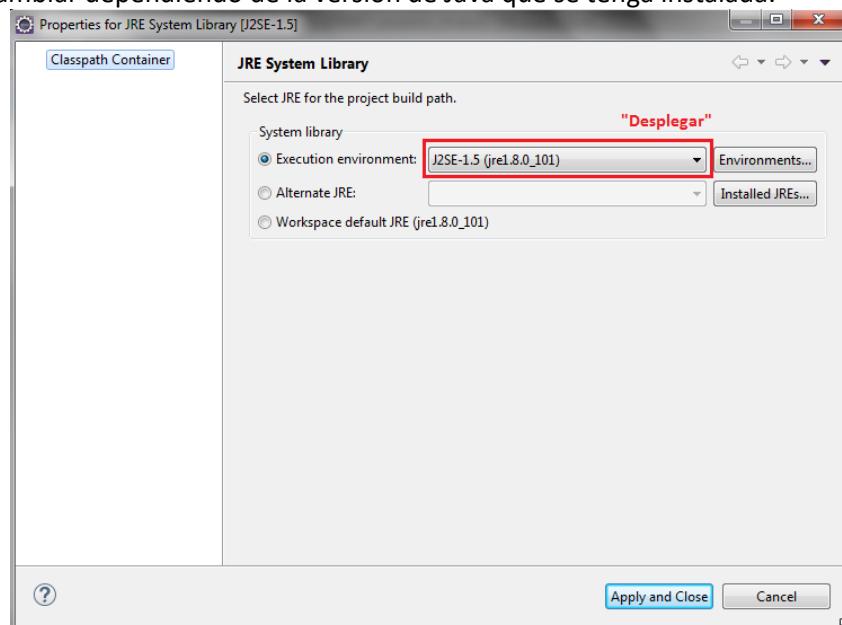


Por cuestiones de practicidad se cambia “JRE System Library” a JSE-1.8:
Click derecho como se indica en la imagen y elegir **Properties**

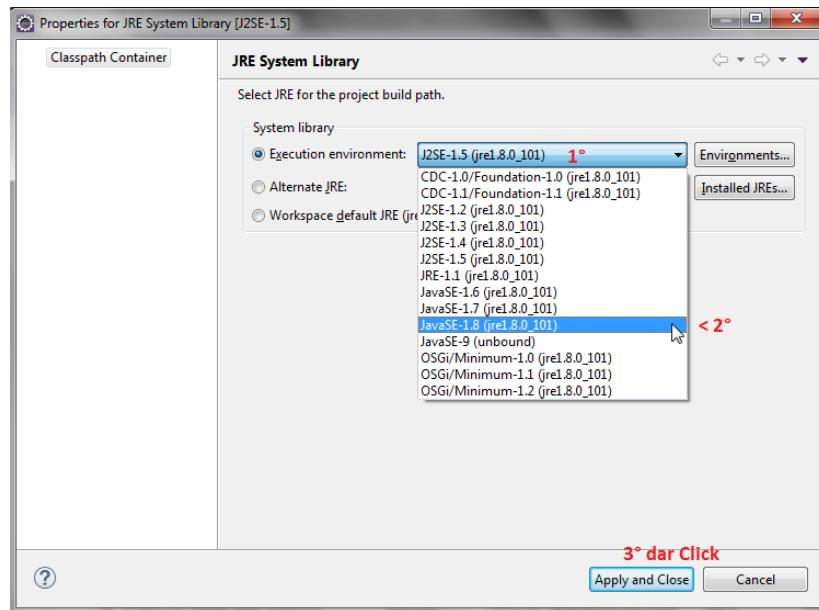


Seleccionar JavaSE-1.8(jre.8.0_144)

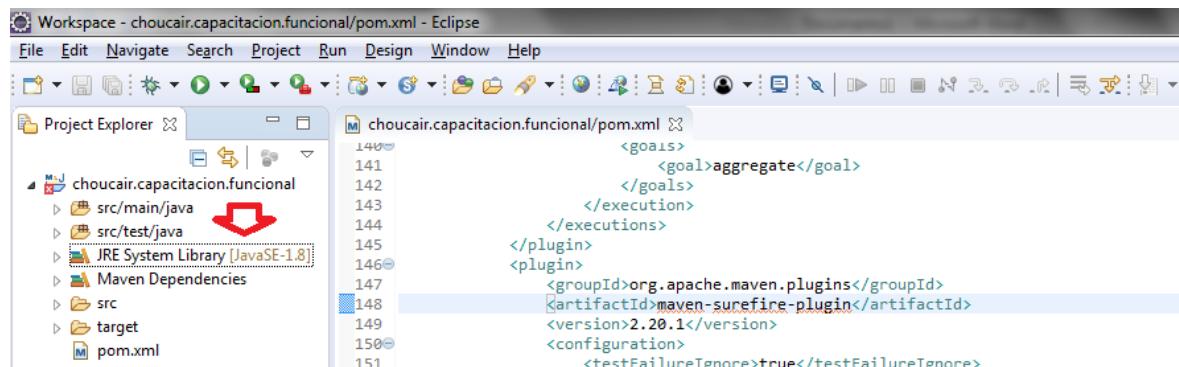
Esto puede cambiar dependiendo de la versión de Java que se tenga instalada.



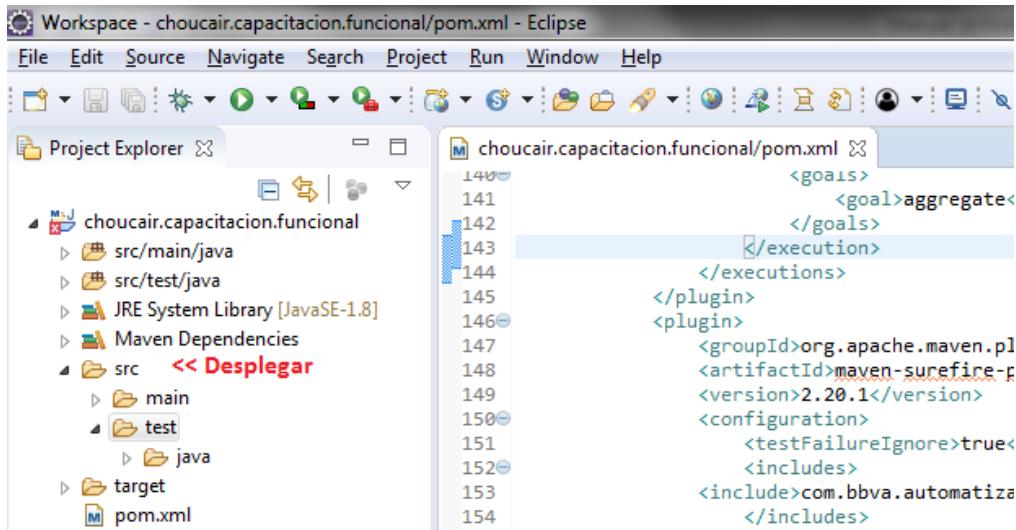
Seleccionar la versión más actualizada instalada



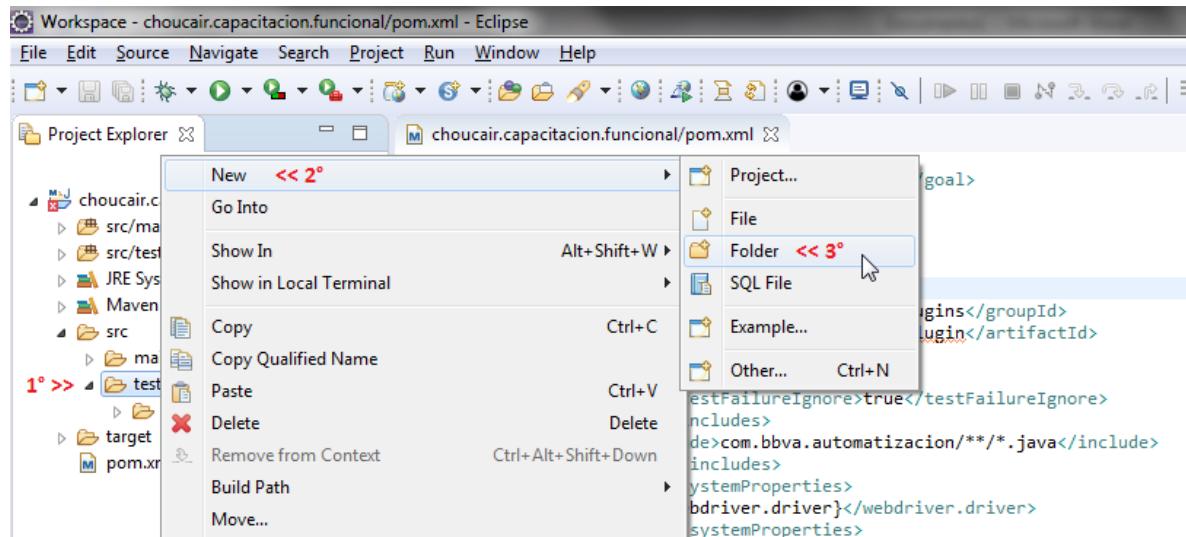
Debe visualizarse como indica la imagen



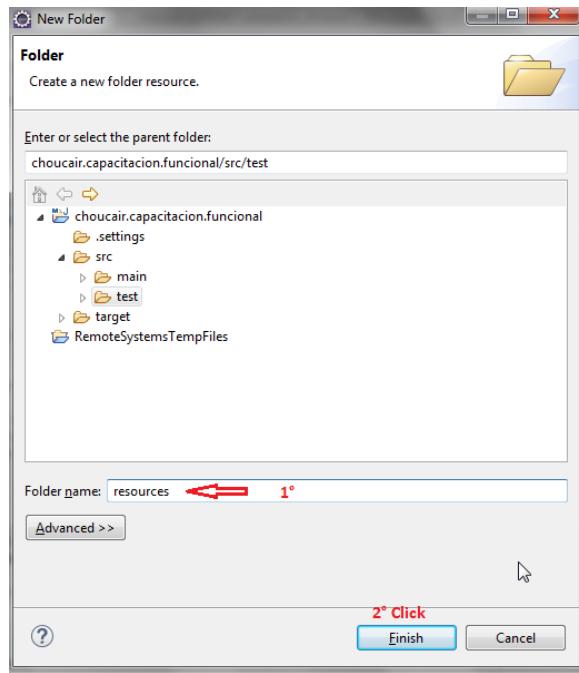
Creación de carpeta “resources” la cual es reconocida por Serenity, donde va a estar todo el proyecto generando los links:



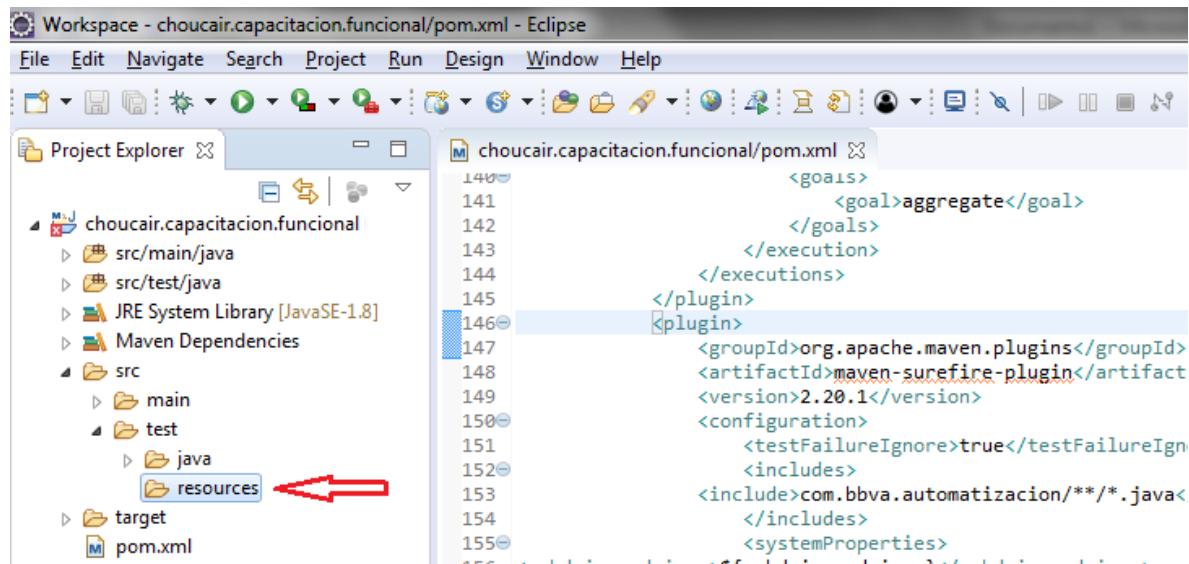
Para crear el directorio, se da click derecho en “test” y se selecciona como indica la imagen



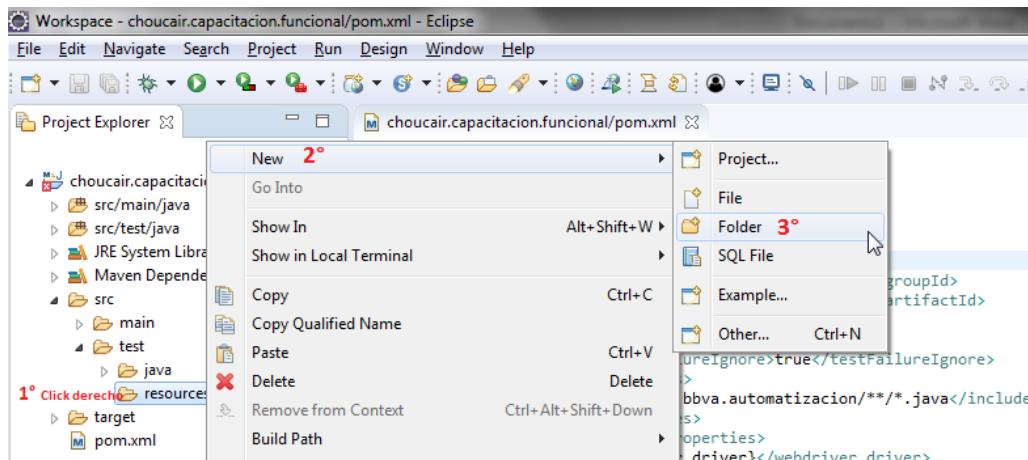
Se le da el nombre al directorio



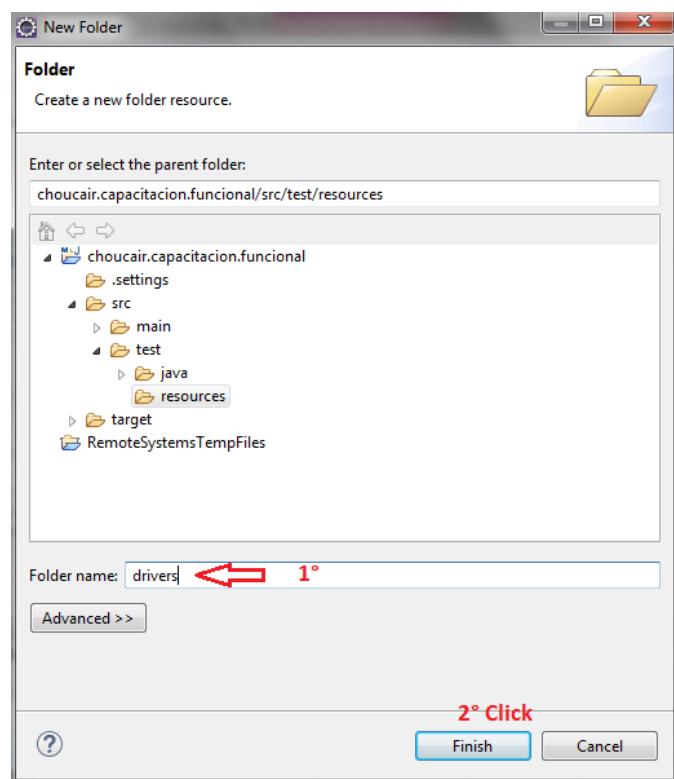
El directorio creado se visualiza como se indica la imagen:



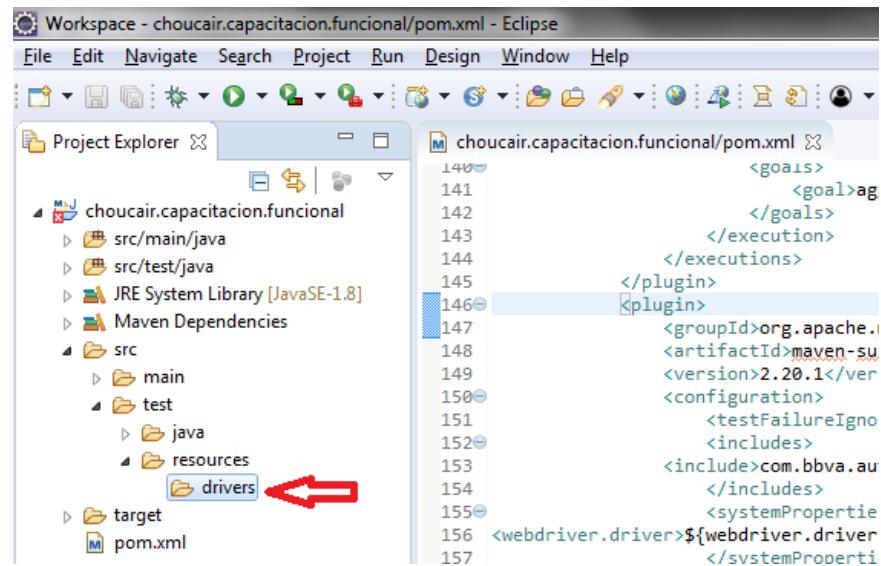
Dentro de la carpeta se crea una con el nombre de “features” donde irán los archivos que se despliegan y la carpeta “drivers”.



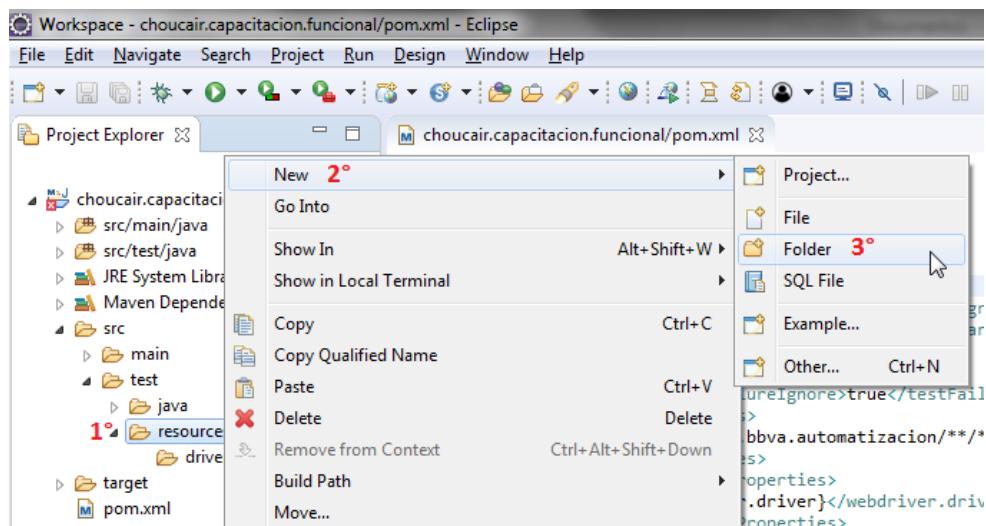
Se abre la ventana para agregar el nombre



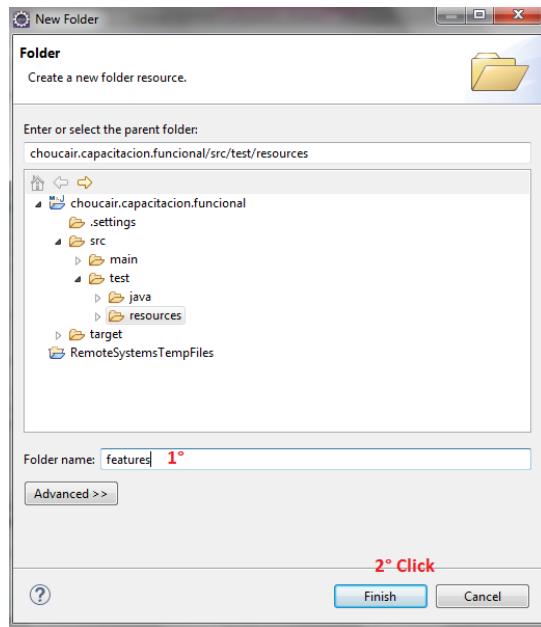
El directorio se visualizará como indica la imagen:



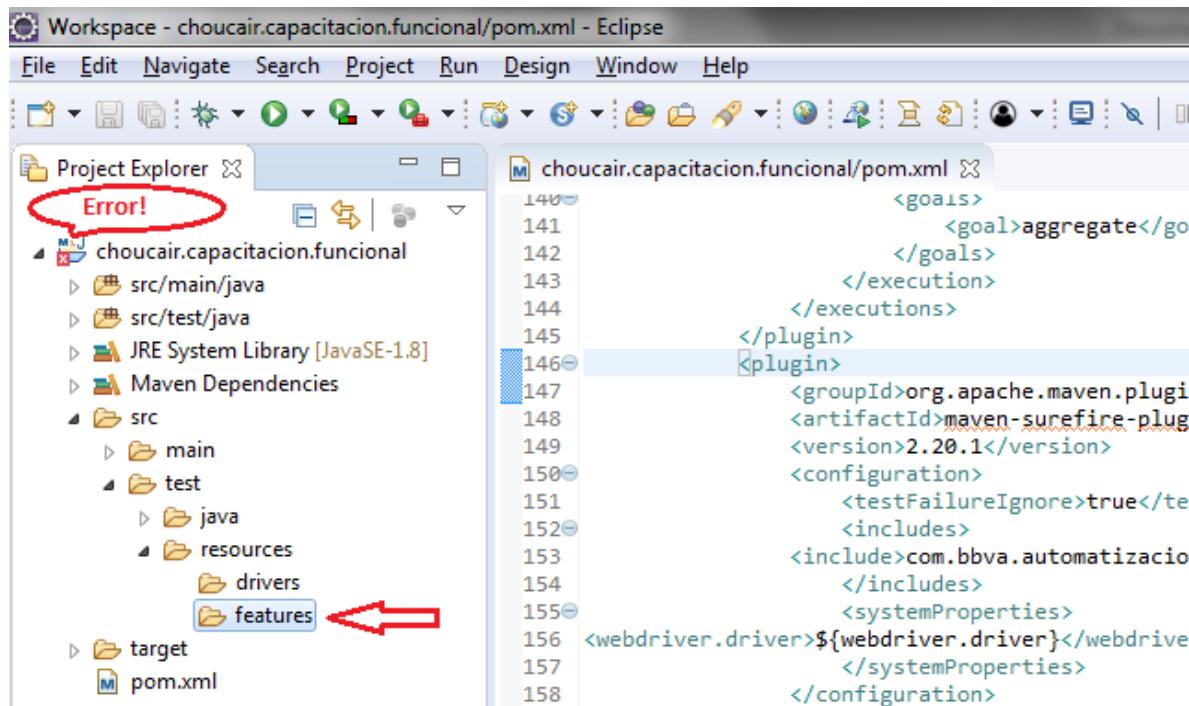
Creación carpeta “features”:



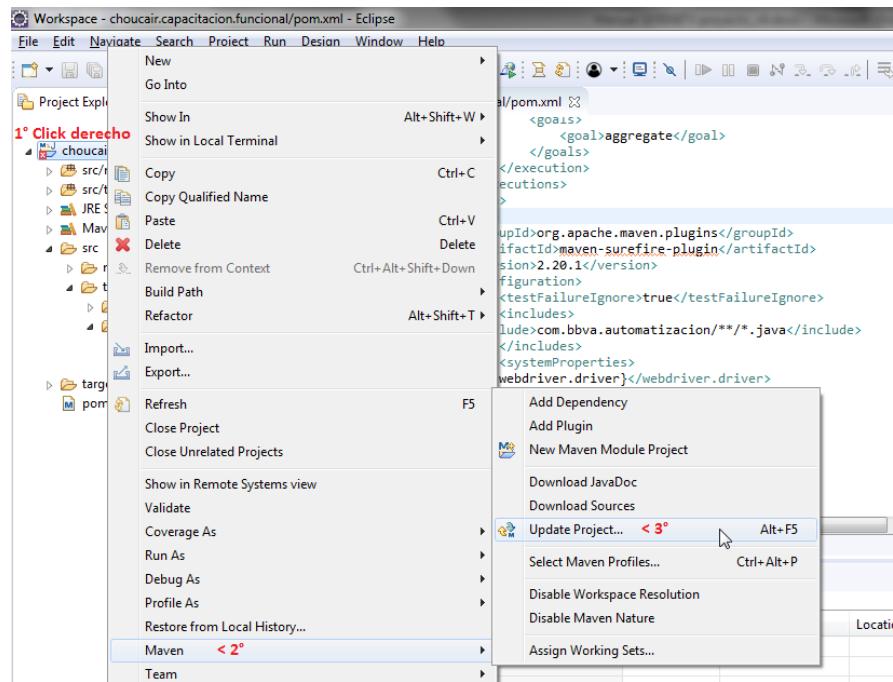
Se da el nombre “**features**” como indica la imagen y se da click en **Finish**



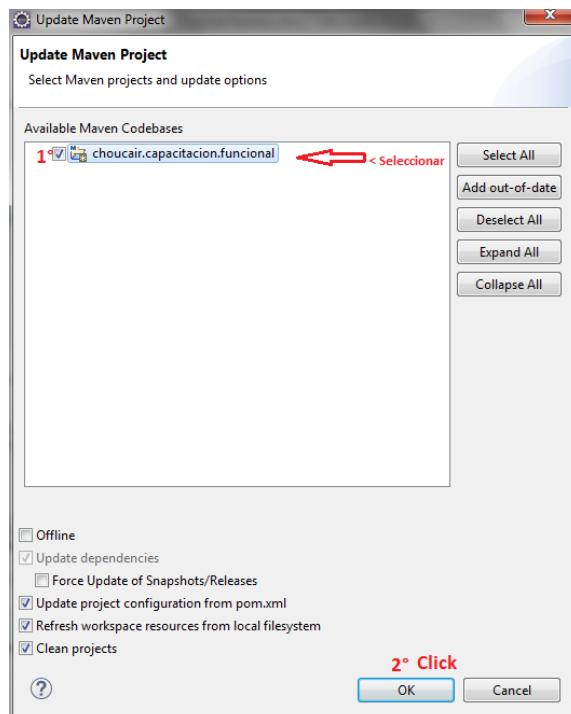
El directorio creado se visualizará como indica la imagen:



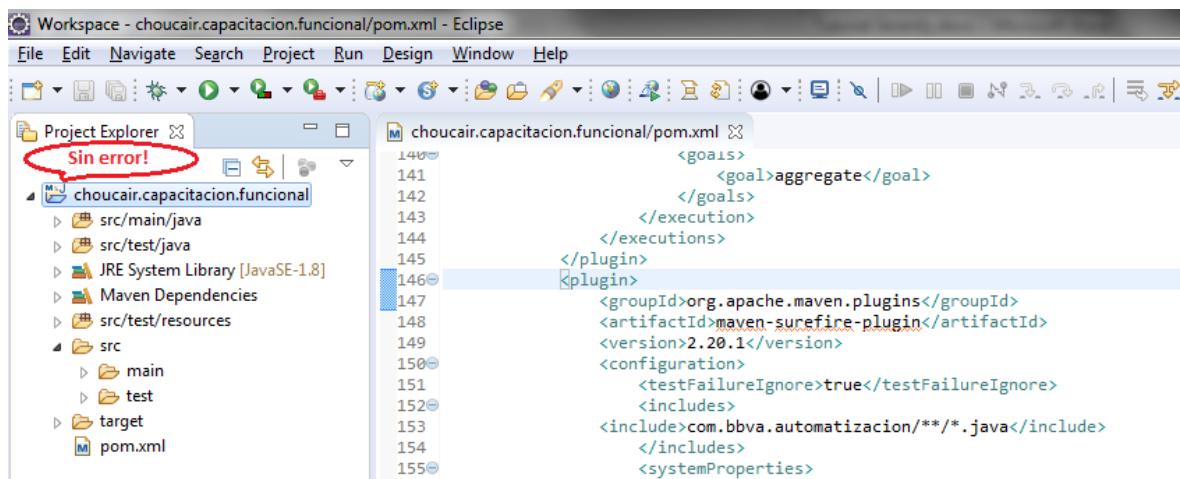
De aquí en adelante se crean las carpetas de aplicativos y módulos que integran éstos, adicional a ello se debe actualizar el proyecto haciendo clic derecho sobre éste siguiendo la siguiente ruta “**Maven\Update Project...**” esta opción corrige el error que se ve en la imagen anterior.



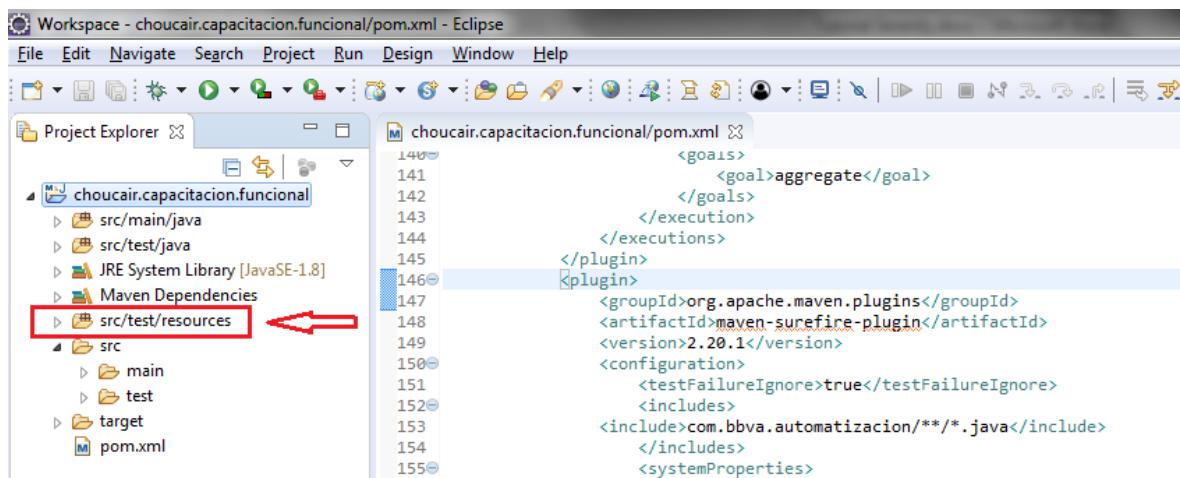
Mostrará la siguiente pantalla:



El error desaparecerá:



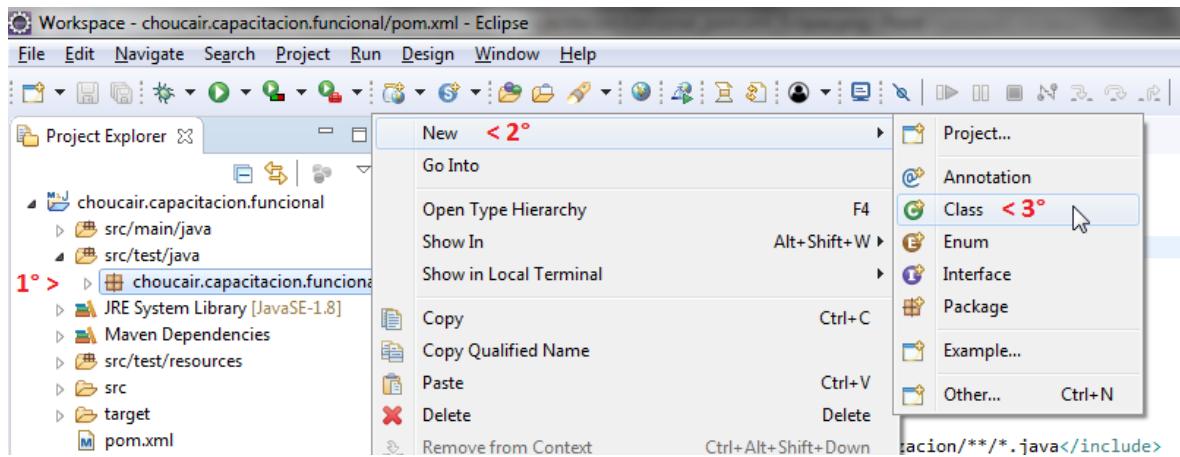
Note que el proyecto creó la carpeta “**src/test/resources**” de forma automática, aunque la creación de las carpetas se hizo en el “**src/test/resources**” de la parte inferior.



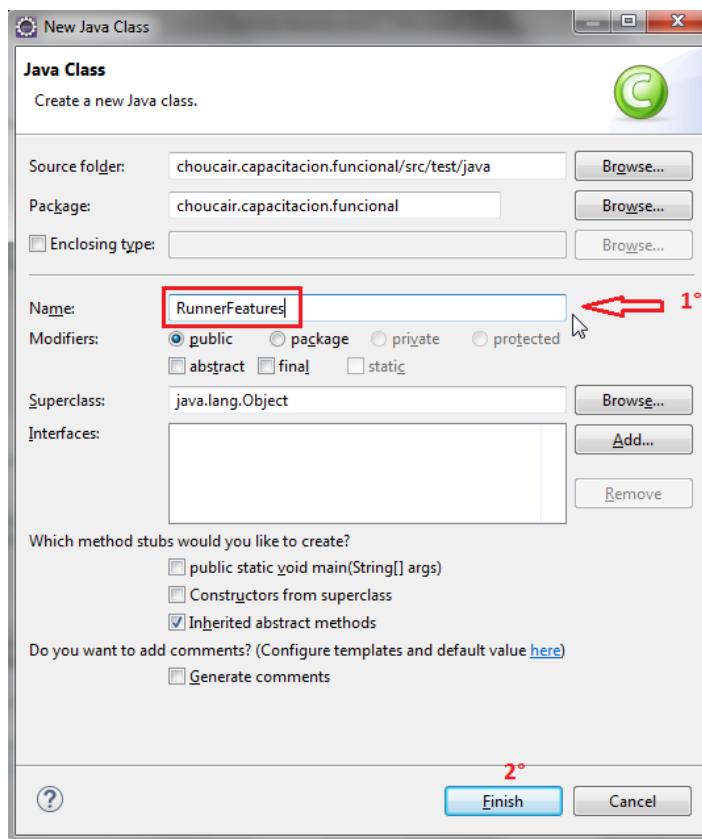
CREACION CLASE “RunnerFeatures”

Para validar el éxito del proceso e integrar con Serenity se crea una clase en java que servirá de prueba inicial como se muestra en la imagen siguiente:

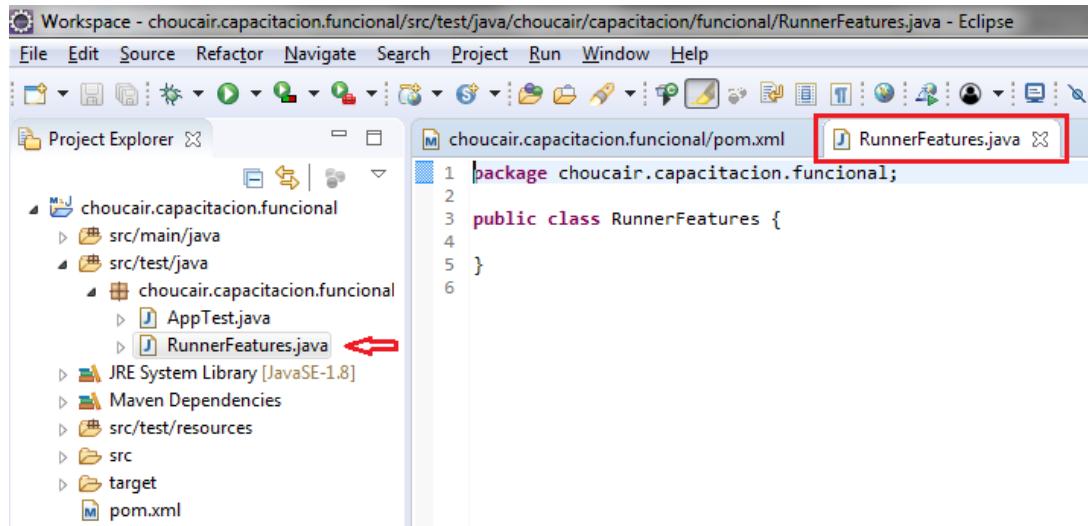
Click derecho en “src/test/java”, seguir la numeración:



Escribir el nombre como indica la imagen “RunnerFeatures”



La clase se visualizará como indica la imagen:

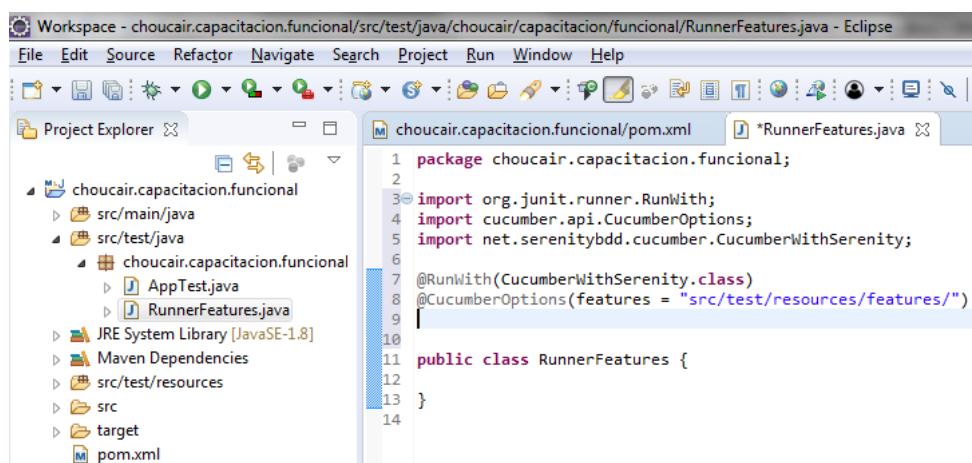


Se hacen los imports requeridos y se indica la ruta donde Serenity encontrará los .features que serán ejecutados:

```
import org.junit.runner.RunWith;
import cucumber.api.CucumberOptions;
import net.serenitybdd.cucumber.CucumberWithSerenity;

@RunWith(CucumberWithSerenity.class)
@CucumberOptions(features = "src/test/resources/features/")
```

Se visualiza como indica la siguiente imagen:

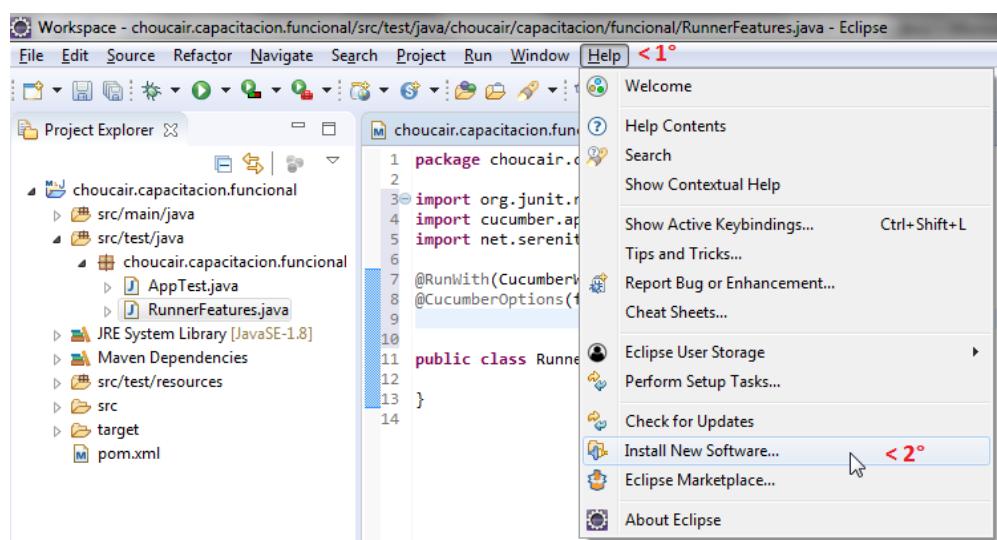


INSTALACIÓN CUCUMBER

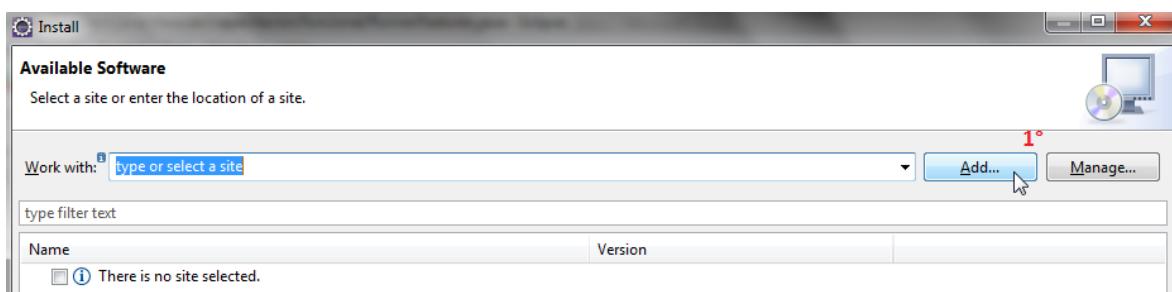
En este punto solo falta instalar cucumber para que eclipse reconozca los archivos *.feature* necesarios en el framework. Para dicha instalación se puede utilizar dos rutas distintas “**Install New Software**” y “**Eclipse Marketplace**” se recomienda usar la primera opción aunque ambas son válidas.

RUTA 1 “**Install New Software**”

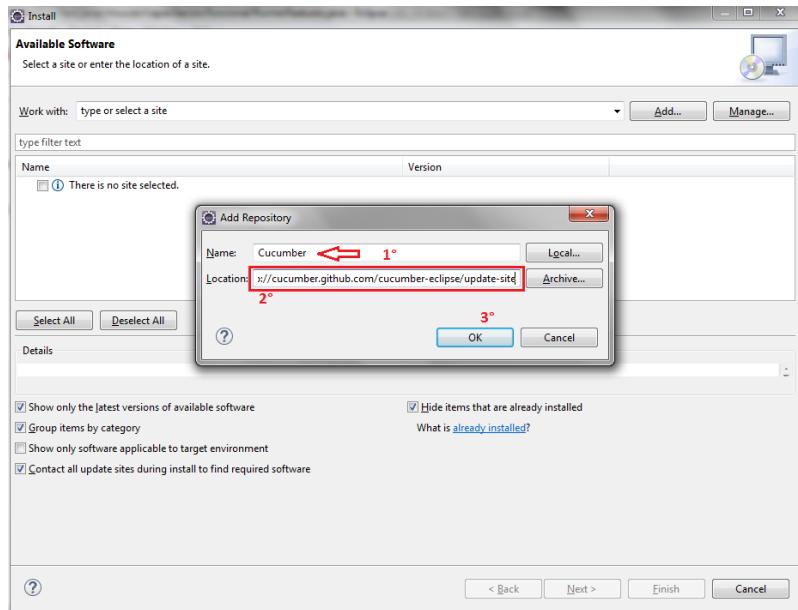
Al dar click sobre la opción “**Help/Install New Software**” se visualiza la pantalla “**Install**” allí se presiona la opción “**Add...**” se procede a ingresar la siguiente información: en el campo “**Name**” Cucumber y en el campo “**Localization**” <http://cucumber.github.com/cucumber-eclipse/update-site> tal como se visualiza a continuación:



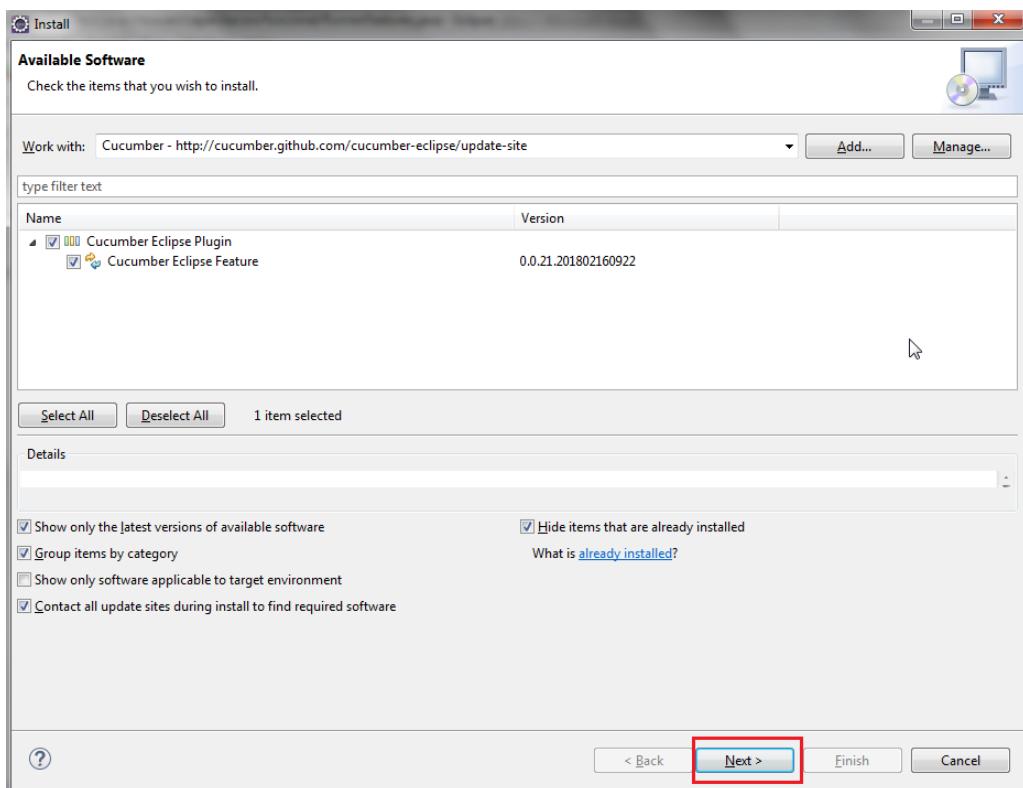
En la pantalla que se visualiza dar click en **Add...**



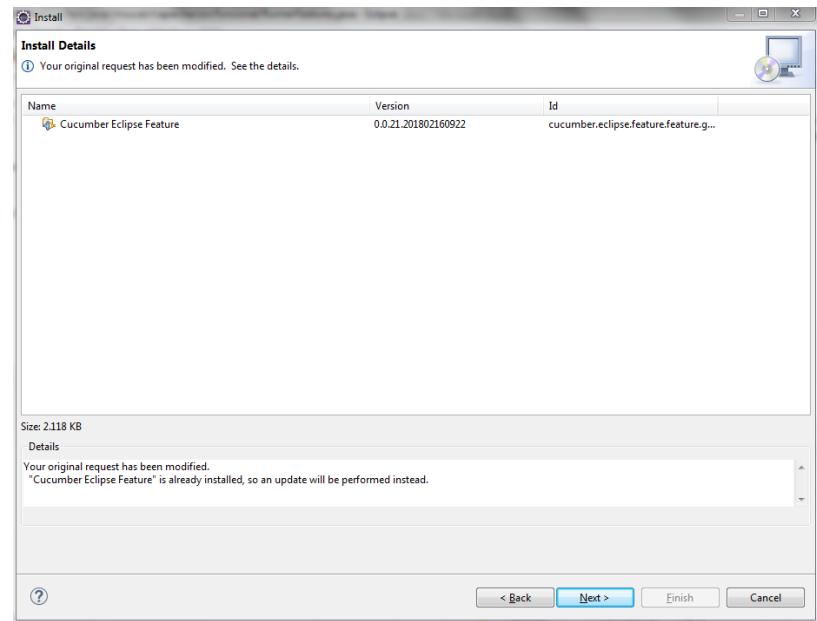
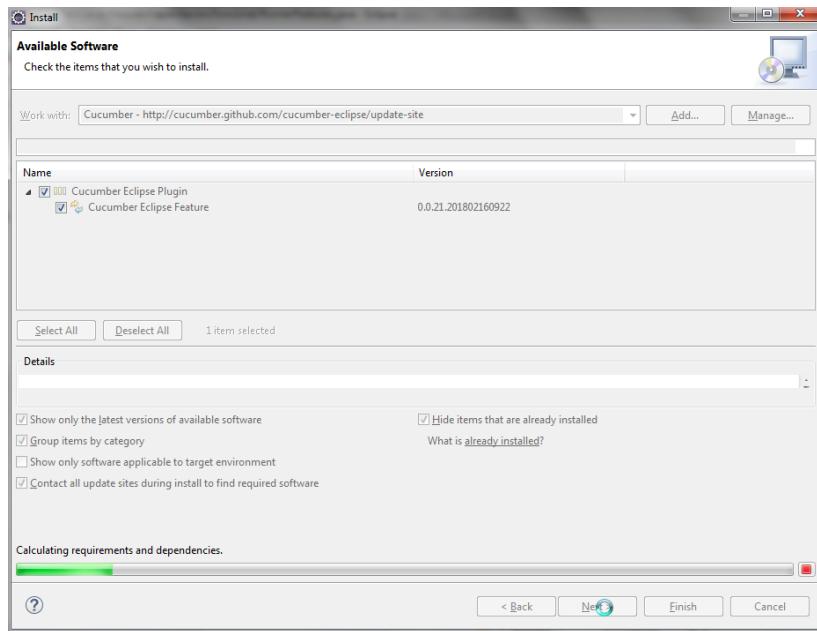
Al dar click sobre el botón “Ok” se muestra la opción de instalar el plugin de cucumber, activamos la casilla y procedemos a la instalación:
<http://cucumber.github.com/cucumber-eclipse/update-site>



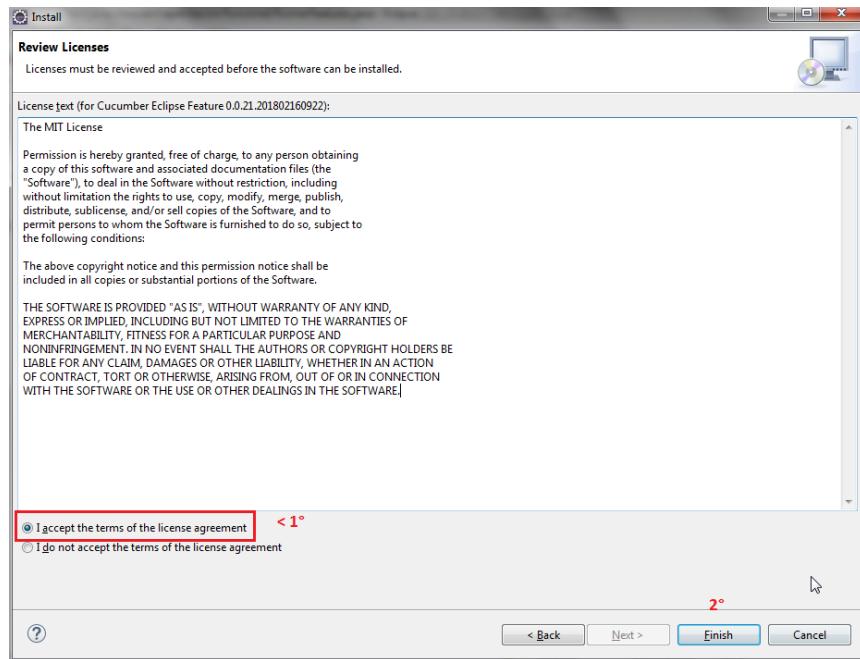
Al dar en la opción “Next” se confirma la instalación y la versión a instalar, si ya se encuentra instalado se procede a actualizar con la versión mostrada:



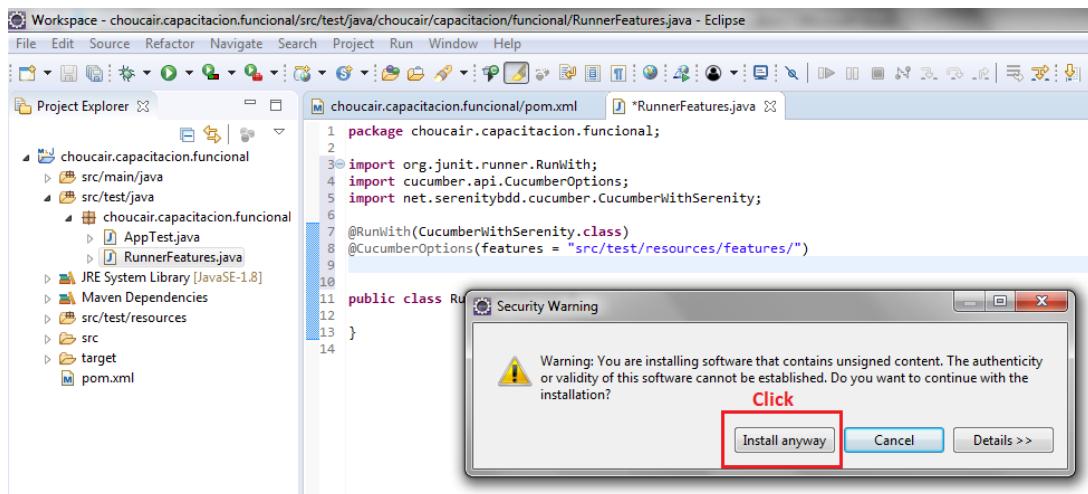
Calculando



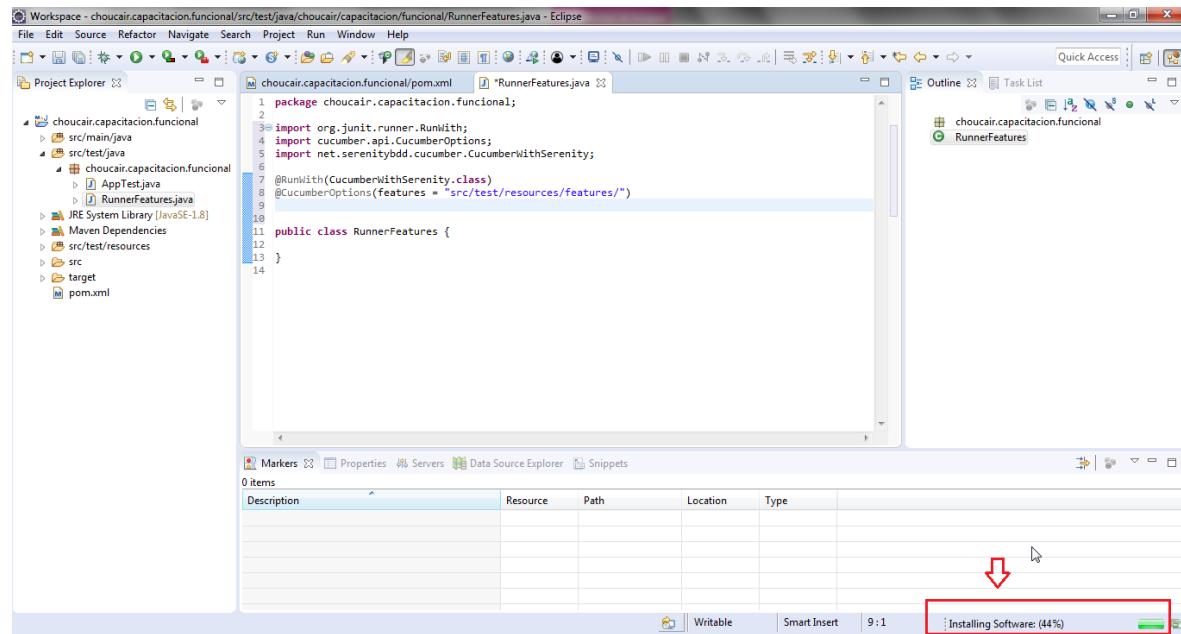
Se aceptan los términos de licencia



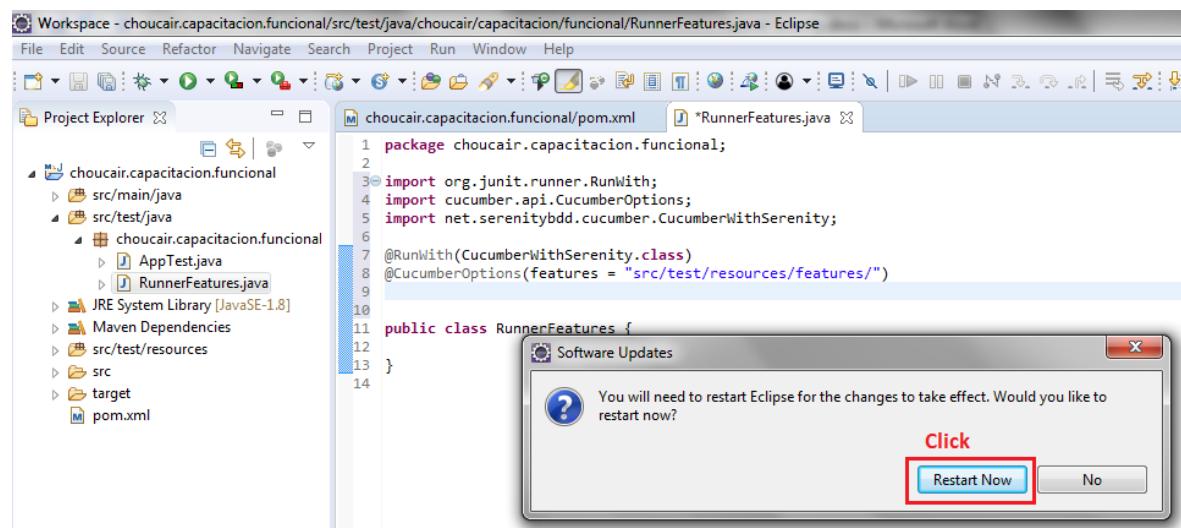
Al finalizar se acepta la ventana de seguridad y su mensaje de advertencia sobre instalación de software:



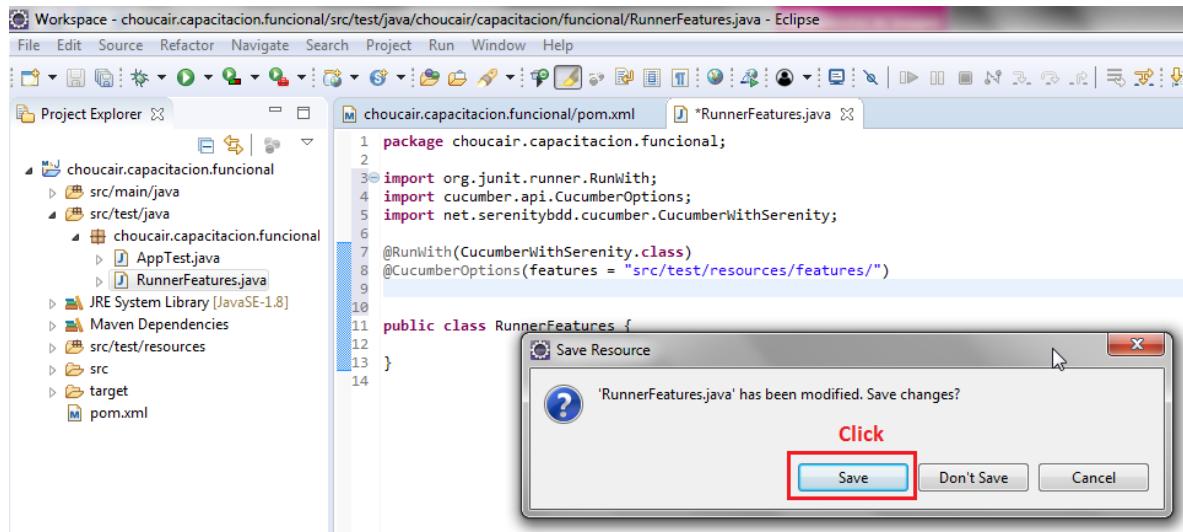
Verificamos en la barra inferior derecha la instalación del software:



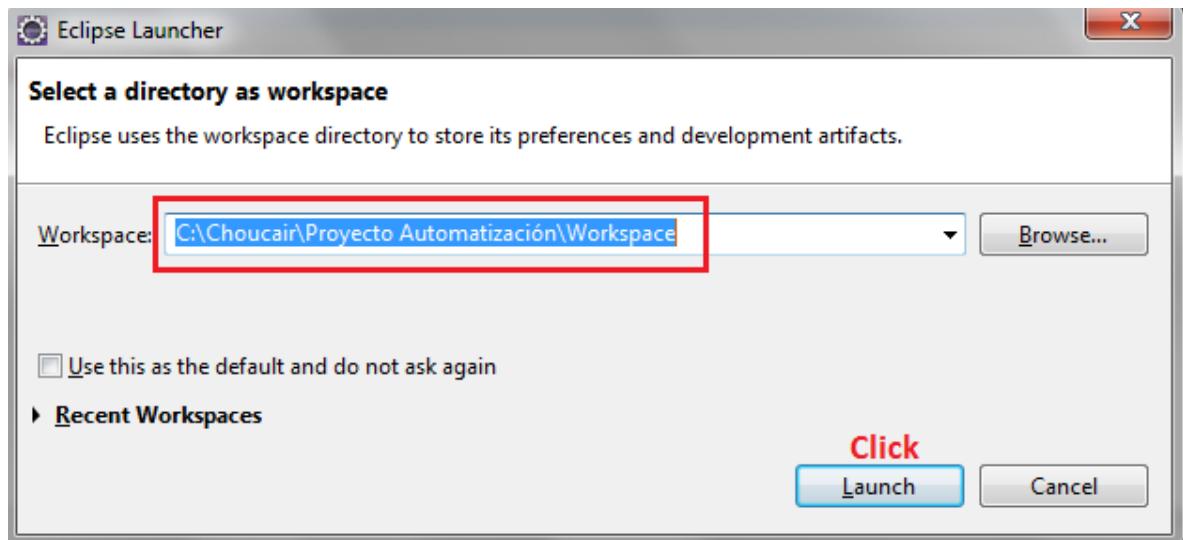
Eclipse pedirá ser restaurado



En caso de haber dado guardar a la clase RunnerFeatures.java, eclipse va a pedir que guarde antes de restaurar



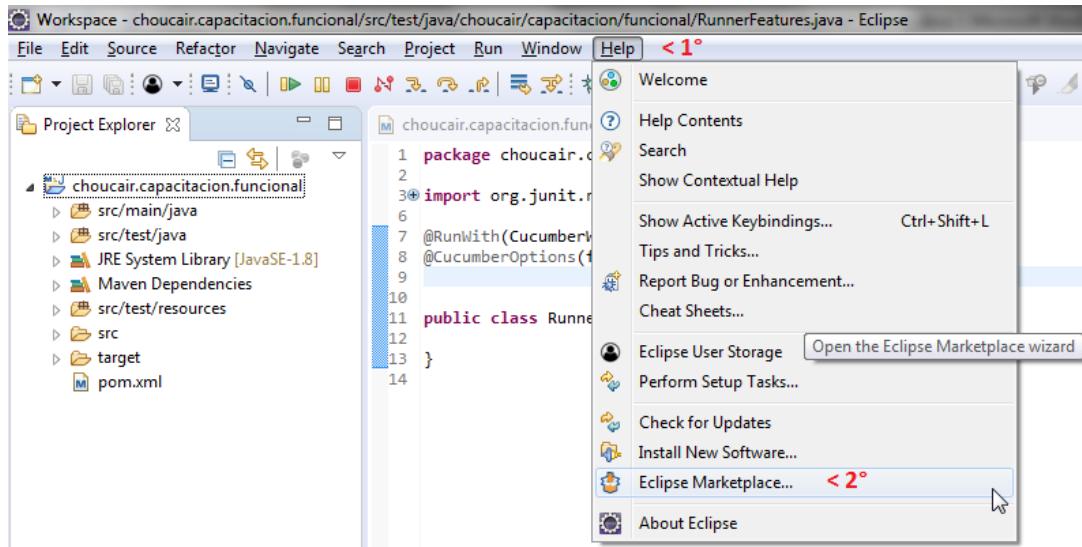
Después de reiniciar se confirma la ruta del workspace



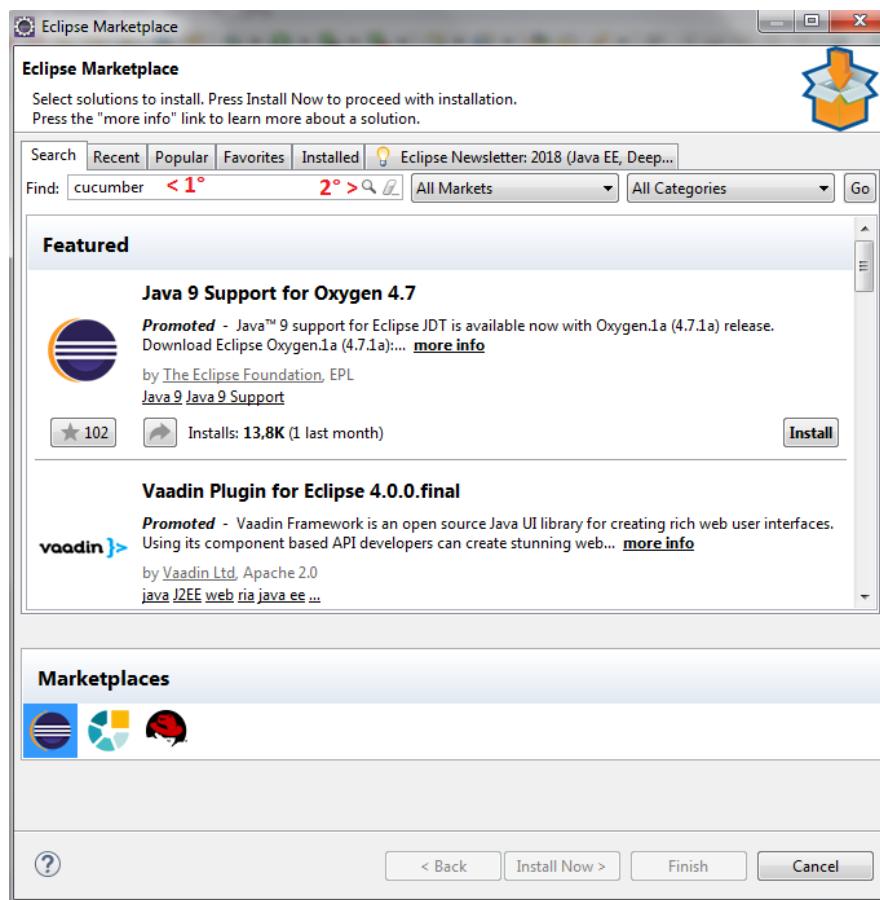
Nota: si se tiene un problema con el plugin, se debe desinstalar y buscar una url válida y actual diferente a la ingresada con anterioridad.

RUTA 2 “Marketplace”: (No hacer si se eligió la RUTA 1)

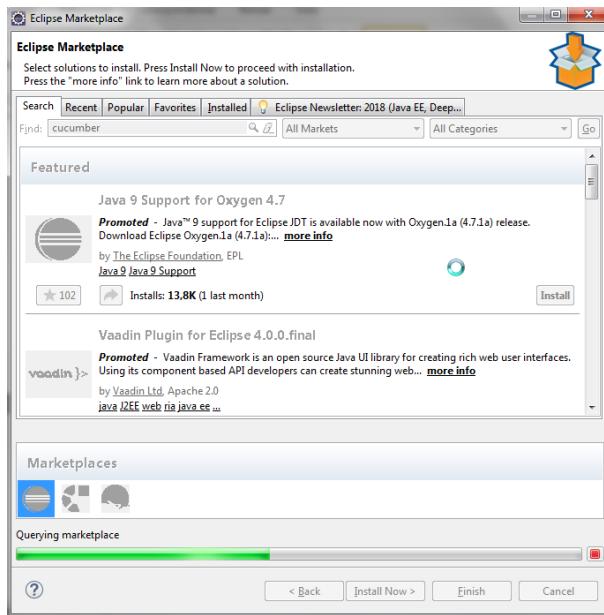
Ir a la barra de herramientas y dar click en la opción **Help “Help/Eclipse Marketplace”**



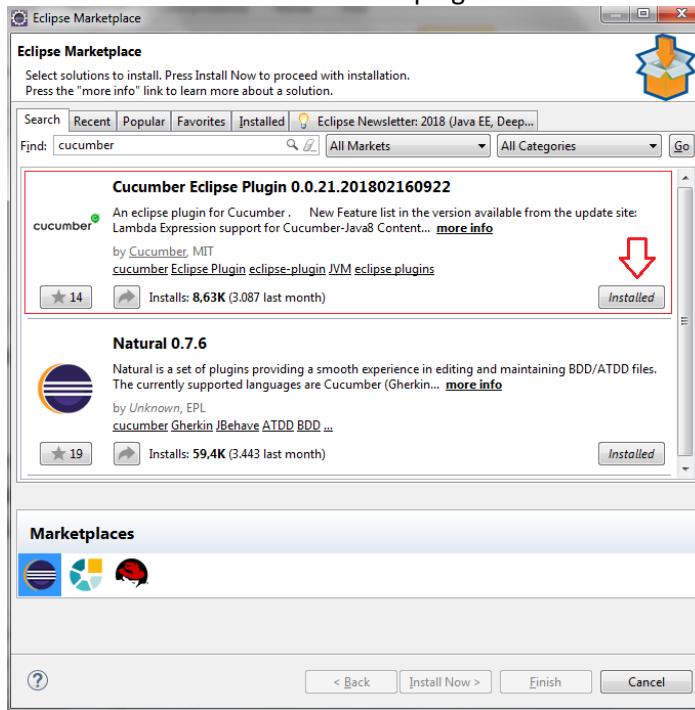
Se escribe el nombre del plugins “cucumber” que se quiere buscar y se da click en la lupa para buscar



Buscando:



Al buscar con la palabra “cucumber” se muestran dos plugins:

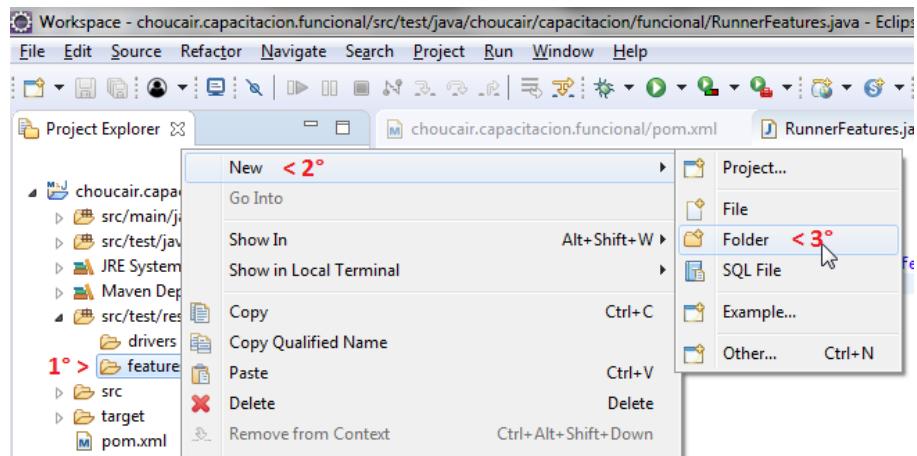


Si se ha realizado la instalación con anterioridad se muestra la opción “**Installed**”, se da click en “**Cancel**”.

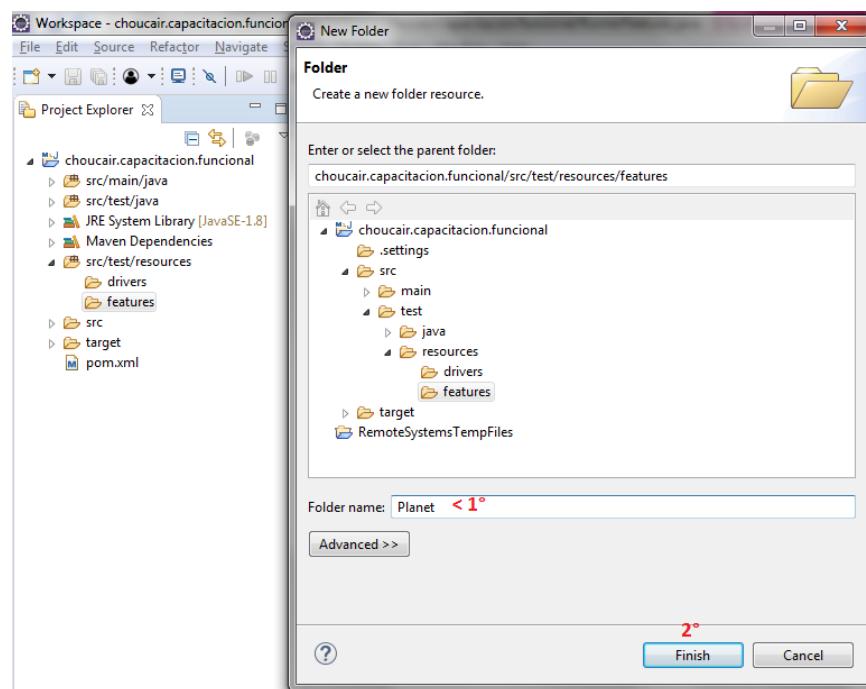
Lo contrario estará presente la opción “**Install**”, dado el último caso se hace click sobre éste y se reinicia eclipse.

CREACIÓN ARCHIVO .feature

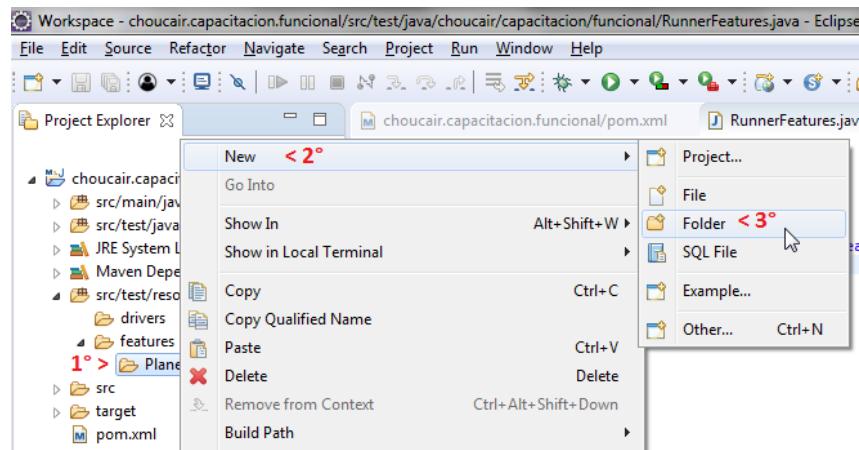
Se crea una carpeta con el nombre del aplicativo que estemos trabajando en el directorio feature:



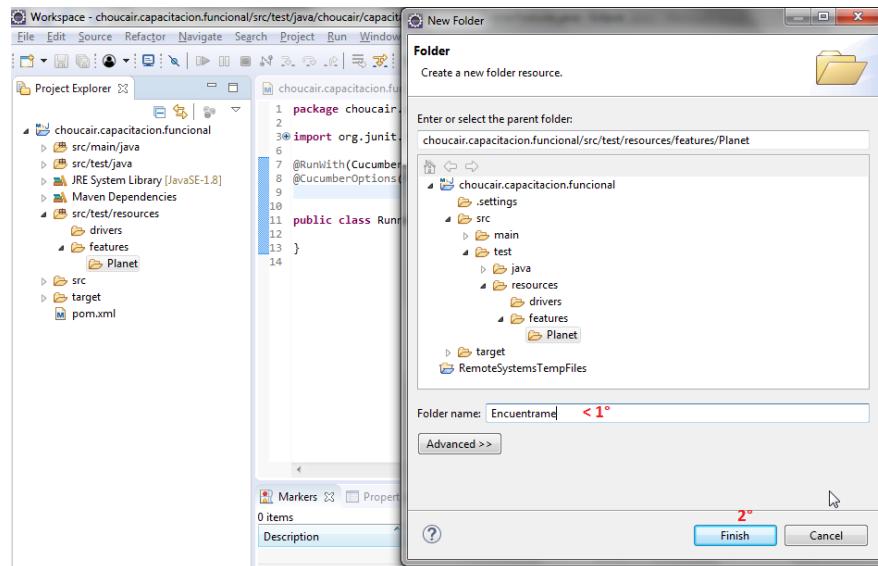
Se ingresa el nombre del aplicativo



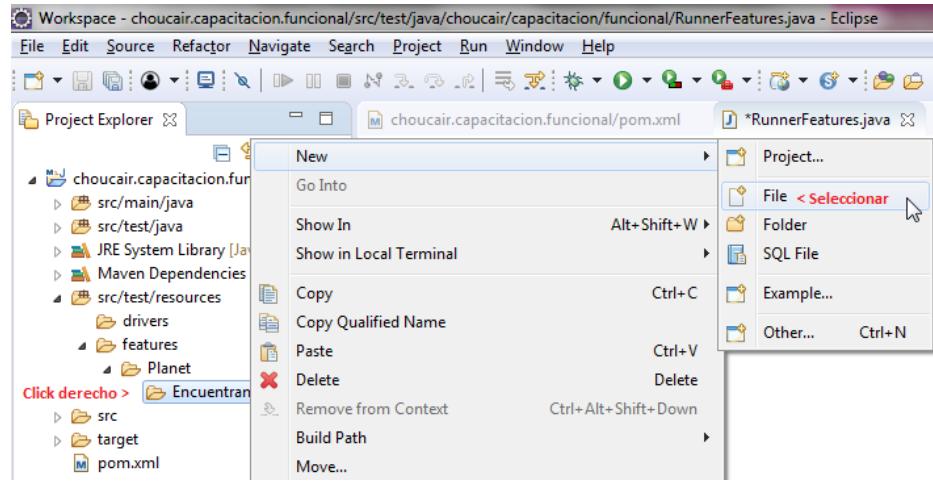
Dentro del directorio del aplicativo se crean los directorios por cada uno de los flujos:



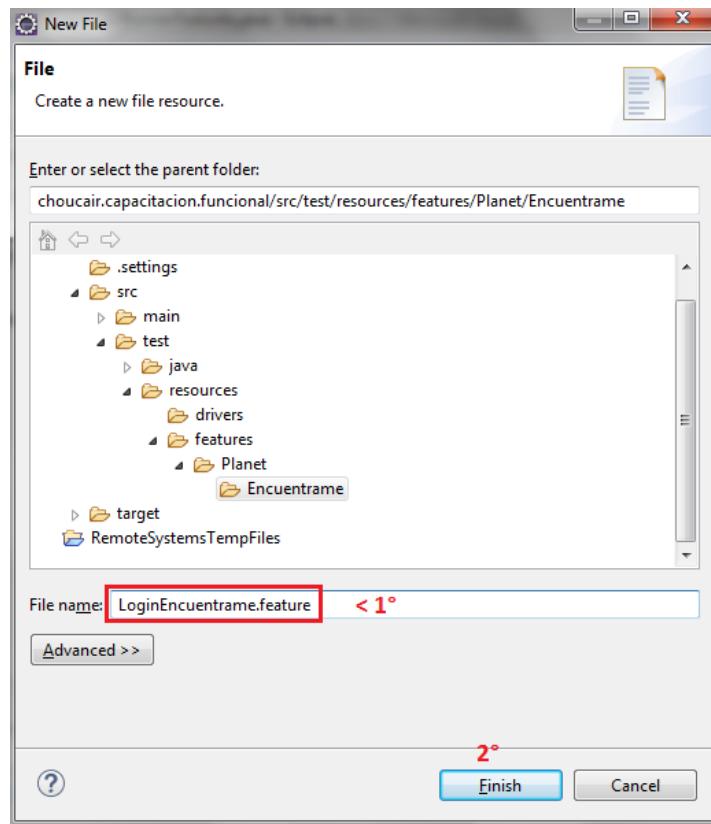
Se ingresa el nombre del directorio donde se creará el archivo



Una vez instalado el plugin de cucumber en eclipse se procede a crear el archivo “.feature” que se ejecute bajo la clase “**RunnerFeature.java**” creada con antelación, para ello se crea un archivo dentro de una de las carpetas finales por ejemplo “**Planet/Encuentrame**”.

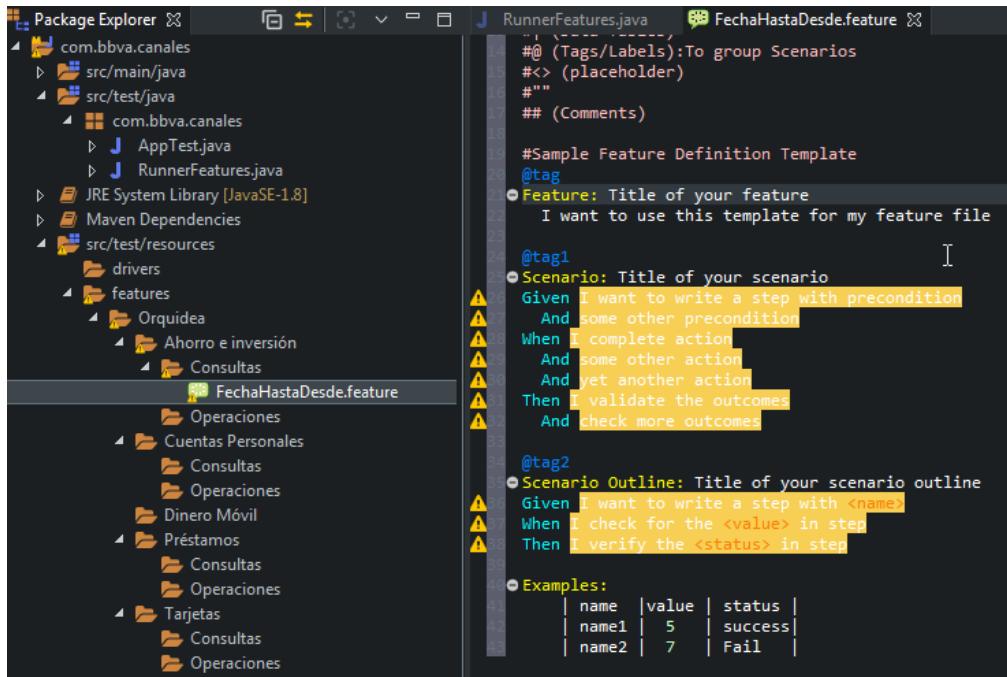


El nombre del archivo debe terminar con “.feature” y ser parte de la operación para dicho módulo siendo tendiente a un caso de prueba funcional, ejemplo:



El archivo .feature es creado y se visualiza como indican las imágenes: Pueden generarse de una de las dos formas, vacío o con estructura guía:

Con estructura GUIA



The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer view displays a project structure under 'com.bbva.canales'. In the center, the code editor shows a file named 'FechaHastaDesde.feature'. The code in the editor is a template for a feature file, starting with '#@ (Tags/Labels): To group Scenarios' and including sections for '@tag', '@tag1', '@tag2', and examples. The code is color-coded with syntax highlighting.

```
RunnerFeatures.java  FechaHastaDesde.feature
-----
#@ (Tags/Labels): To group Scenarios
#<> (placeholder)
#"
## (Comments)

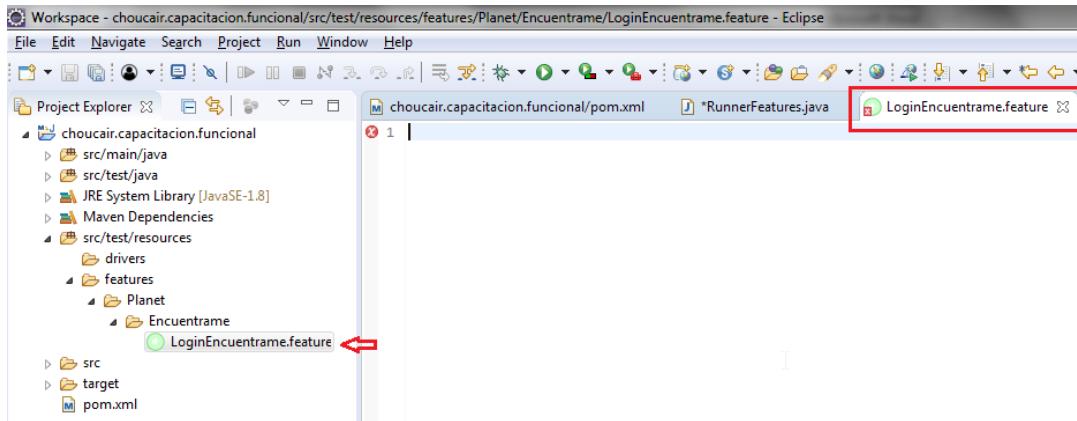
#Sample Feature Definition Template
@tag
● Feature: Title of your feature
  I want to use this template for my feature file

@tag1
● Scenario: Title of your scenario
  Given I want to write a step with precondition
    And some other precondition
  When I complete action
    And some other action
    And yet another action
  Then I validate the outcomes
    And check more outcomes

@tag2
● Scenario Outline: Title of your scenario outline
  Given I want to write a step with <name>
  When I check for the <value> in step
  Then I verify the <status> in step

● Examples:
  | name | value | status |
  | name1 | 5 | success|
  | name2 | 7 | Fail |
```

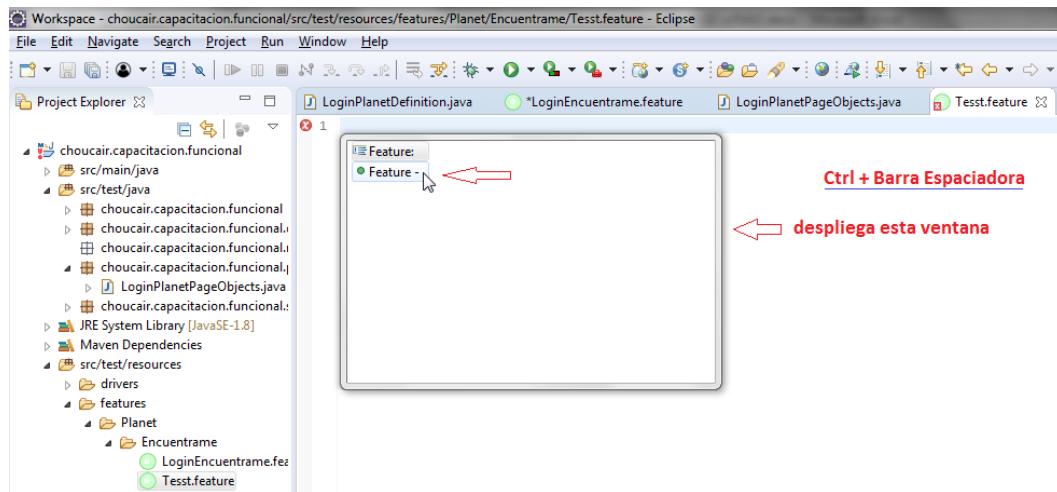
Con estructura VACIA



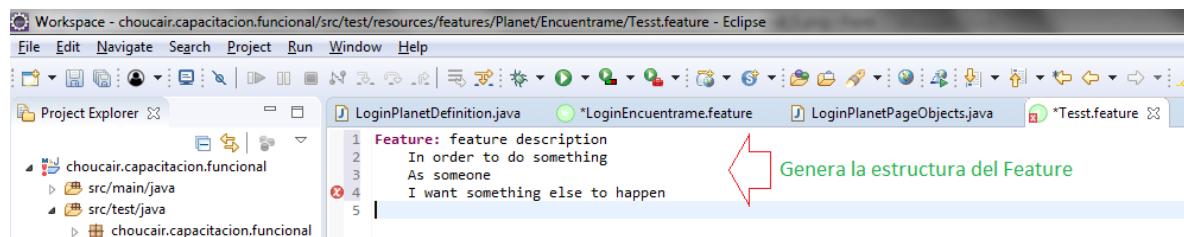
CONSTRUCCIÓN DEL ARCHIVO “.feature”

Cuando inicia vacía se debe llenar como indica la imagen ejemplo:

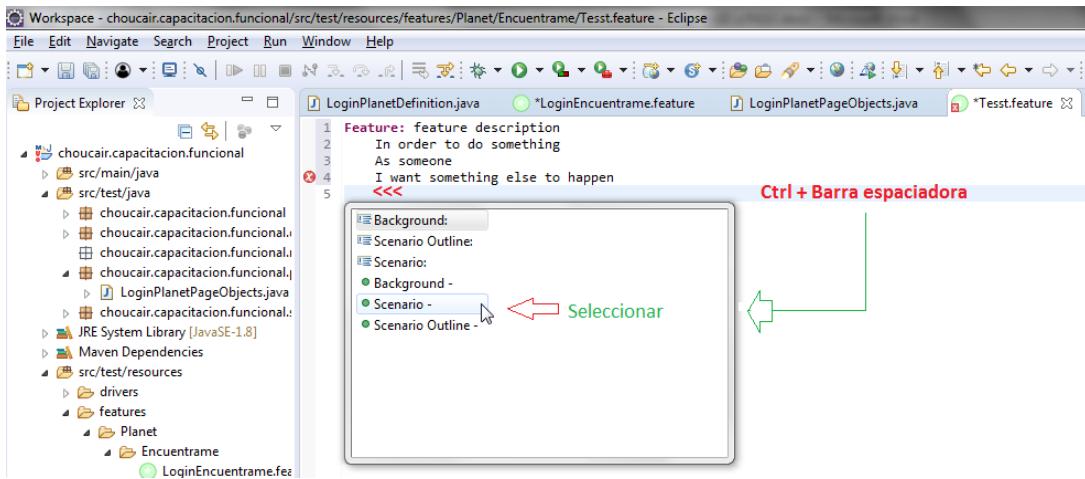
Dar click dentro del archivo .feature y pulsar **Ctrl + Barra espaciadora**, lo cual desplegará la pantalla emergente que muestra la siguiente imagen, donde se debe seleccionar **Feature -**



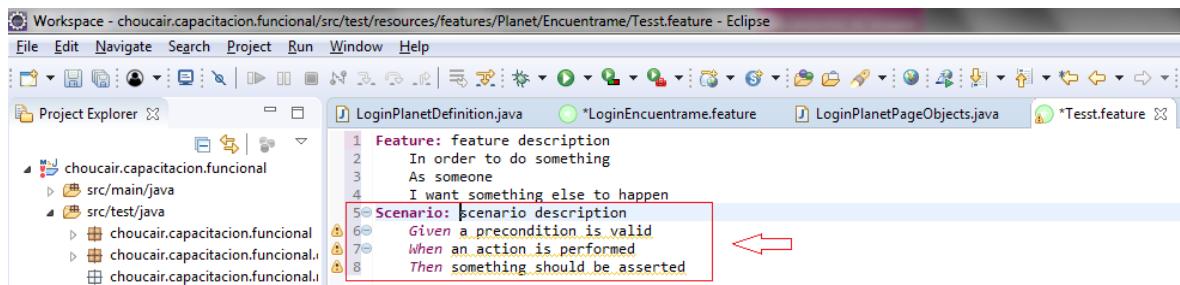
Después de seleccionar se genera la estructura como indica la siguiente imagen:



Se vuelve a presionar **Ctrl + Barra espaciadora**, lo cual desplegará la pantalla emergente que muestra la siguiente imagen, donde se debe seleccionar **Scenario** -



Se genera la estructura del Scenario:



Creadas las estructuras se diligencian según la funcionalidad a automatizar, en la siguiente imagen es un ejemplo:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "choucair.capacitacion.funcional".
- Editor:** Displays the file `LoginPlanetDefinition.java` containing Gherkin steps and Java code.
- Annotations:**
 - A red arrow points to the line `Given Que tengo mis datos de ingreso` with the text "**< Para ejecutar 1 caso**".
 - A green arrow points to the line `When Ingreso el usuario "gvargas"` with the text "**< Para ejecutar varios casos**".
 - A red arrow points to the table in line 24 with the text "**Cada línea es un caso de prueba.**" and a green note below it "**"Agregar comillas a cada parametro"**".

```

1 ## Creado por Gustavo Vargas
2 ## 19-02-2018
3 Feature: Login Planet
4   Para ingresar al aplicativo Planet
5   Como un usuario logueado
6   Quiero hacer alguna actividad dentro del aplicativo
7
8
9@ Scenario: Logueo en planet con usuario valido
10@ Given Que tengo mis datos de ingreso
11@ When Ingreso el usuario "gvargas"
12@ And Ingreso la contraseña "Choucair.2017"
13@ Then Verifico que ingresé al aplicativo
14
15
16@ Scenario Outline: Logueo en planet con usuario valido
17@ Given Que tengo mis datos de ingreso
18@ When Ingreso el usuario <txtNombre>
19@ And Ingreso la contraseña <txtClave>
20@ Then Verifico que ingresé al aplicativo
21
22@ Examples:
23@ | txtNombre | txtClave |
24@ | "gvargas" | "Choucair.2017" |
25@ | "gvargas" | "Choucair.2017" |
26

```

Cuando se crea un **Scenario Outline**: como en la imagen anterior una alternativa es agregar comillas a los parámetros que se envían en cada caso de prueba.

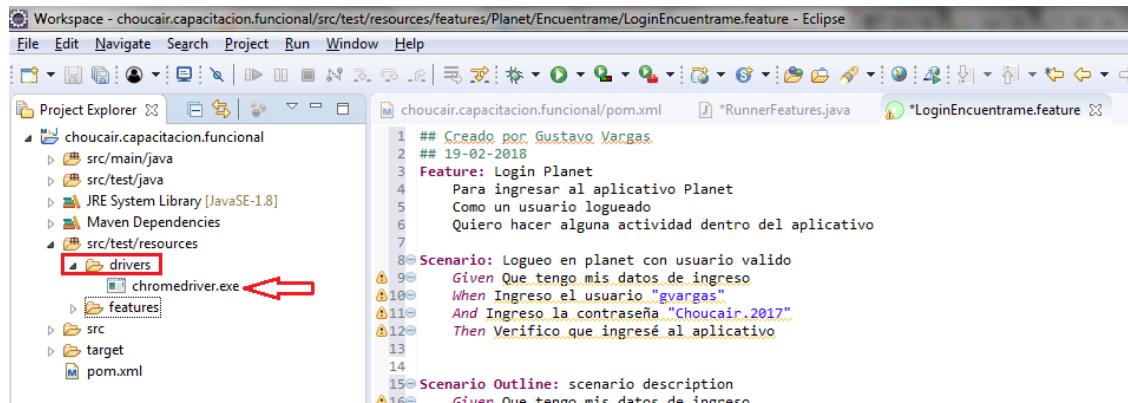
DESCARGA E INSTALACIÓN DE DRIVERS

Para ChromeDriver la URL: <https://sites.google.com/a/chromium.org/chromedriver/downloads>

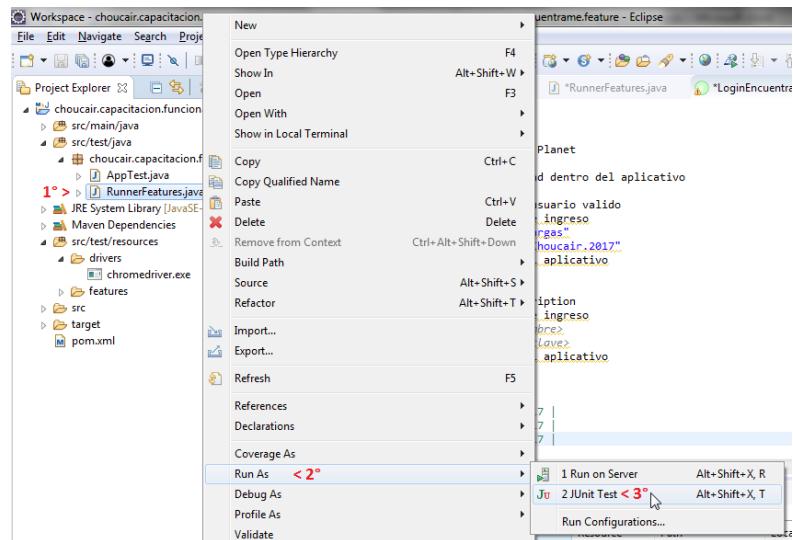
The screenshot shows the official ChromeDriver download page:

- Header:** Shows the URL <https://sites.google.com/a/chromium.org/chromedriver/downloads>.
- Title:** ChromeDriver - WebDriver for Chrome
- Left Sidebar:** Includes links for CHROMEDRIVER, CAPABILITIES & CHROMEOPTIONS, CHROME EXTENSIONS, CHROMEDRIVER CANARY, CONTRIBUTING, DOWNLOADS, and GETTING STARTED.
- Right Sidebar:** Shows the latest release information: "Latest Release: ChromeDriver 2.35" with a download link and a note "Supports Chrome v62-64".
- Bottom:** Includes a "Changes include:" section and a search bar.

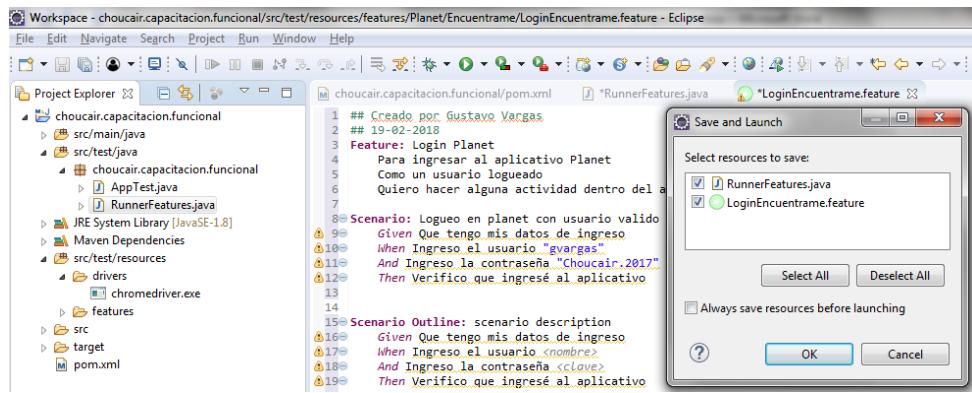
Descargado el archivo se copia y se pega en la carpeta “drivers”



Se verifica que no exista error ejecutando desde la clase de java creada (**RunnerFeatures**), con antelación con JUnit Test:

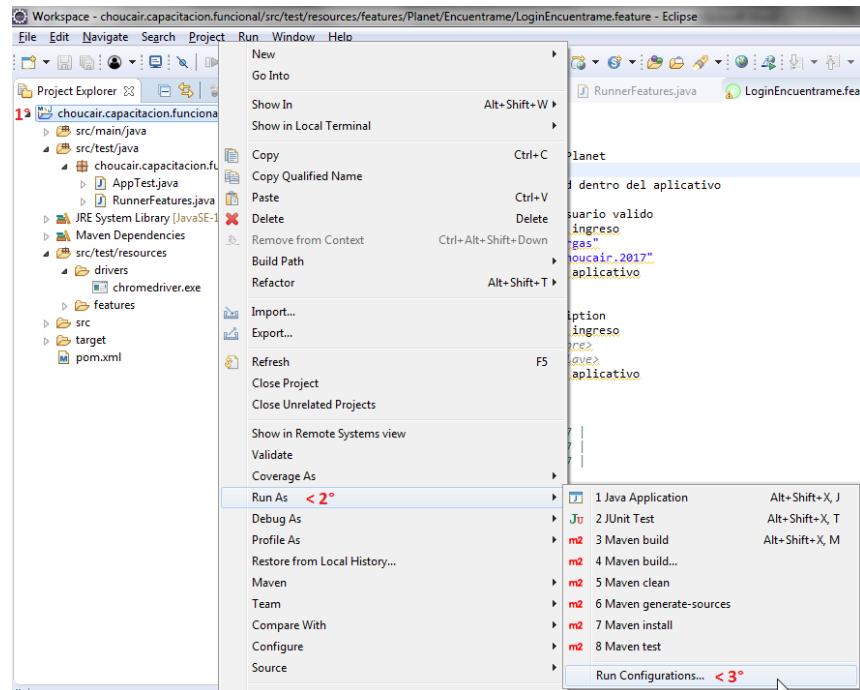


Si hay archivos por guardar mostrará una venta indicando cuales son para seleccionar:

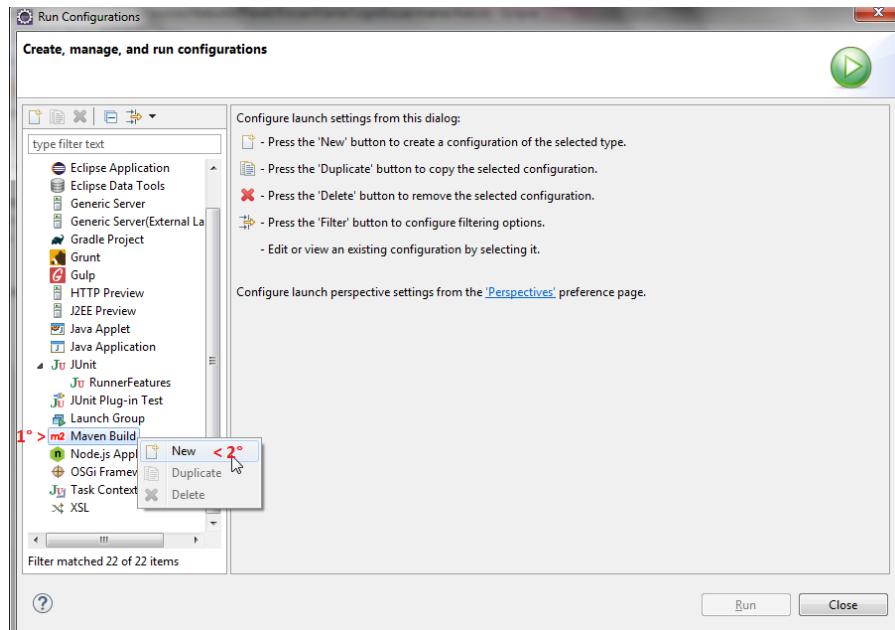


CONFIGURACIÓN DE MAVEN INTEGRANDO SERENITY

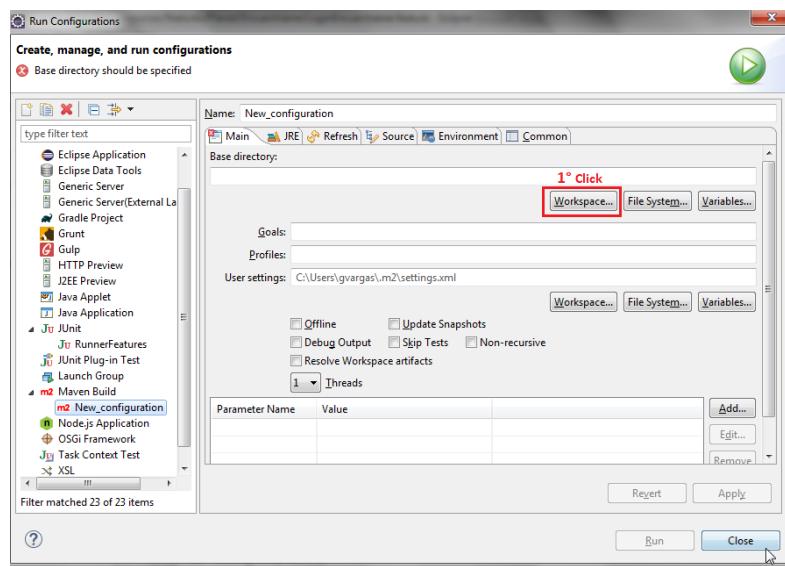
Esto se hace, haciendo clic derecho sobre el proyecto hasta “Run As” y seguidamente “Run Configuration”:



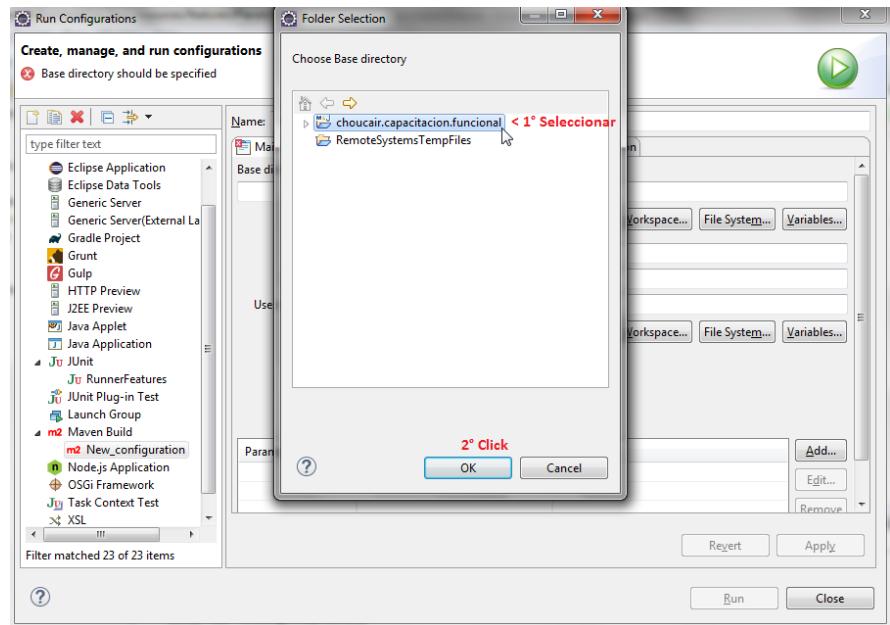
En la pantalla que se activa, seguir los pasos que indica la imagen siguiente:



Seleccionar el Workspace: Click en el botón que indica la siguiente imagen



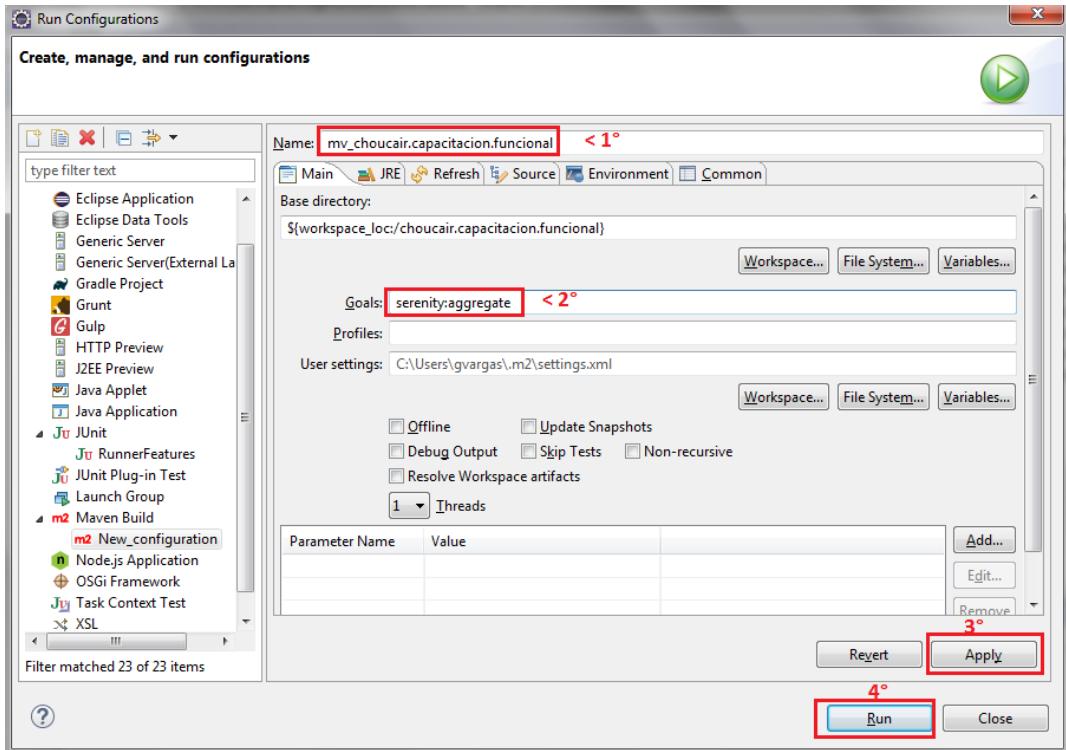
Seleccionar el directorio base



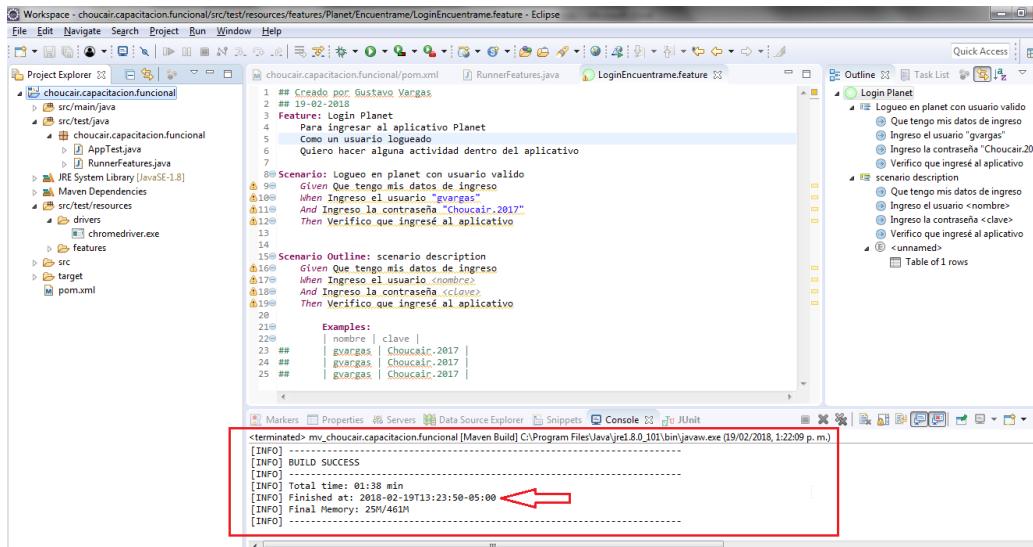
En Name: colocar el nombre del proyecto precedido del mv_ , como se muestra en la imagen:

En la opción “Goals” se ingresa: “serenity:aggregate”:

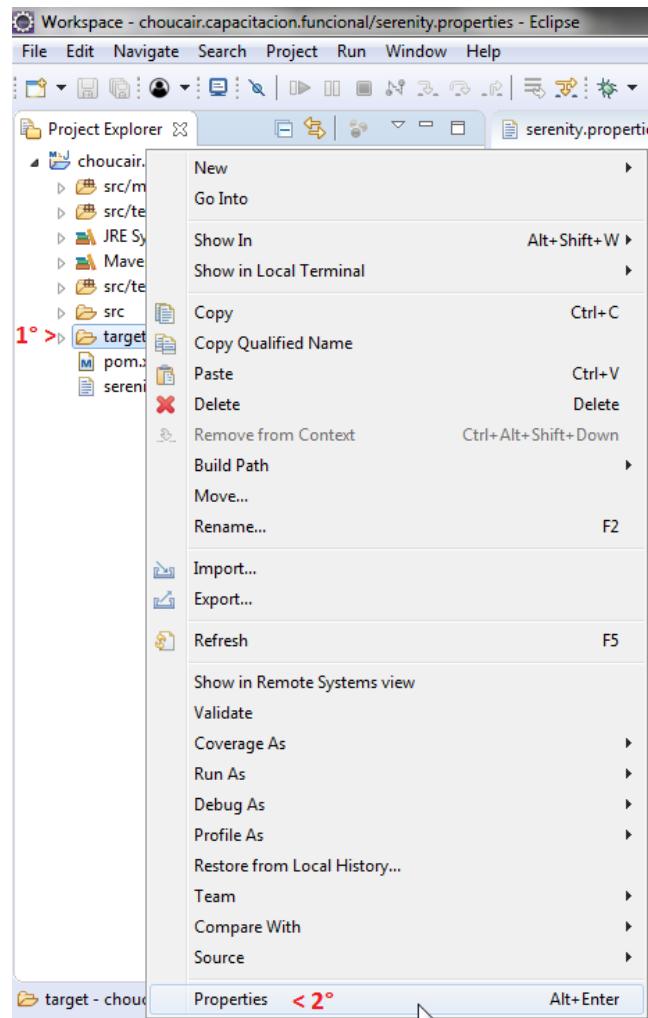
Se presiona la opción “Apply” y “Run” y se verifica la ejecución del mismo:



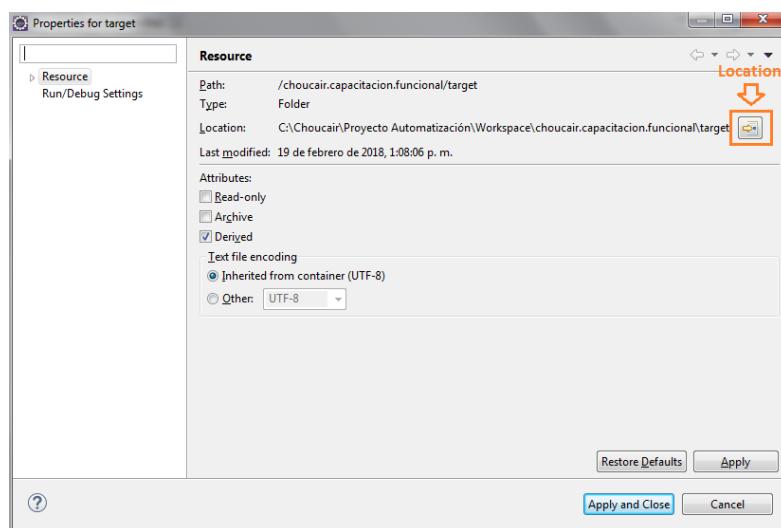
La actualización termina cuando se muestra el mensaje que indica la siguiente imagen



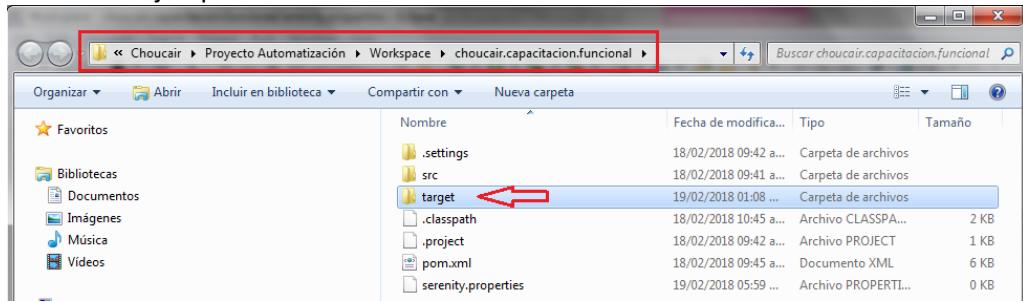
Para revisar la estructura de las evidencias debemos hacer clic derecho sobre las propiedades de la carpeta “target” en la opción “Properties” luego damos clic en el ícono de “Location”



Se da click donde indica la imagen:

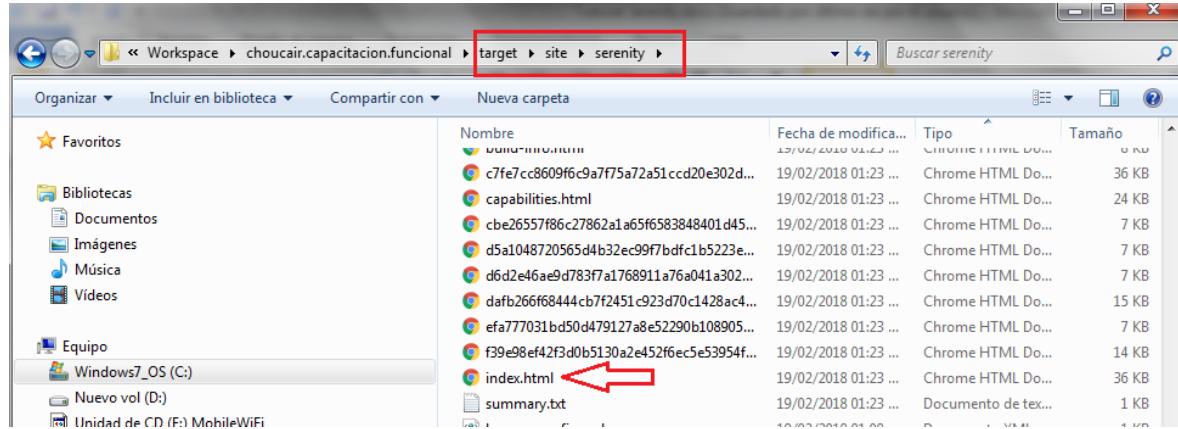


De ésta forma se abrirá la carpeta donde se almacenan las evidencias del proyecto, la imagen que sigue muestra un ejemplo de dicha ruta:



Se debe ingresar hasta la ruta que indica la imagen [/site/Serenity/index.html](#):

Allí se busca el “**index.html**” de la evidencia y se verifica las rutas creadas por **Serenity** hasta el **feature** creado:



The screenshot shows the Serenity Reports interface with the following details:

- Title: Serenity Reports
- URL: file:///C:/ChoucairBBVA/Workspace/com.bbva.canales/target/site/serenity/index.html
- Header: Serenity BDD
- Navigation: Home, Overall Test Results (checked), Requirements, Themes, Capabilities, Features
- Report Generated: 24-10-2017 18:20
- Section: Test Results: All Tests
- Content:
 - 0 test scenarios
 - Distribution of tests (including rows in data-driven tests) by test result:
 - Test Count
 - Weighted Tests
 - Show/Hide Pie Chart

Aplicativo:

Overall Test Results Requirements Themes Capabilities Features Report generated 24-10-2017 18:20

Themes (1)

Show 25 entries Search:

ID	Theme	Capabilities	Automated Tests	Manual Tests	Test Results	Working Capabilities
	+ Orquidea	1	0	0	0% 0%	0%

Módulo:

Test Type Total Pass Fail Pending Ignore/Skip
Automated 0 0 (0%) 0 (0%) 0 (0%) 0 (0%)
Manual 0 0 (0%) 0 (0%) 0 (0%) 0 (0%)
Total 0 0 (0%) 0 (0%) 0 (0%) 0 (0%)
Total Duration 0 seconds

Capabilities (1)

Show 100 entries Search:

ID	Capability	Features	Automated Tests	Manual Tests	Test Results	Working Features
	+ Ahorro e inversión	1	0	0	0% 0%	0%

Showing 1 to 1 of 1 entries Previous 1 Next

Funcionalidad:

Total 0 0 (0%) 0 (0%) 0 (0%) 0 (0%)
Total Duration 0 seconds

Features (1)

Show 100 entries Search:

ID	Feature	Features	Automated Tests	Manual Tests	Test Results	Working Features
	+ Consultas	1	0	0	0% 0%	0%

Showing 1 to 1 of 1 entries Previous 1 Next

Caso de prueba:

Home > Requirements > Consultas > Fecha Hasta Desde

Overall Test Results Requirements Themes Capabilities Features Report generated

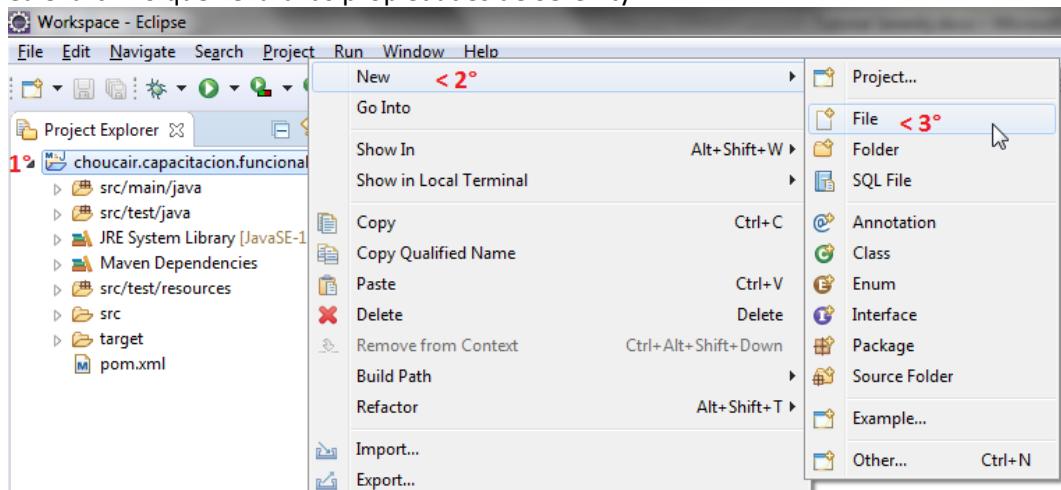
Feature: Fecha Hasta Desde

Fecha hasta desde

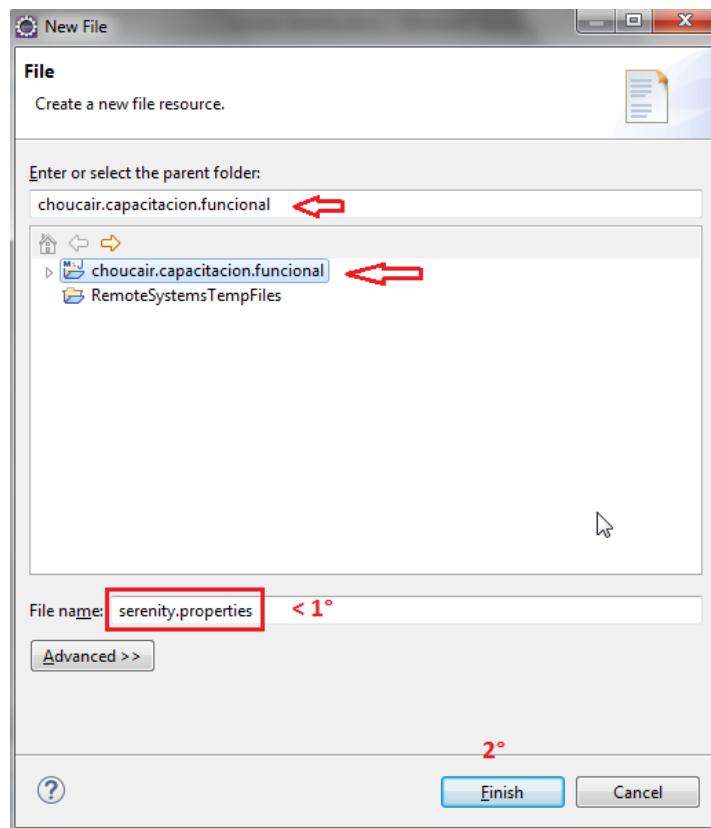
Nótese como el **.feature** tenía el nombre de “**FechaHastaDesde.feature**” y en el caso se muestran las palabras separadas para una mejor lectura de la prueba.

CREACIÓN ARCHIVO PROPIEDADES

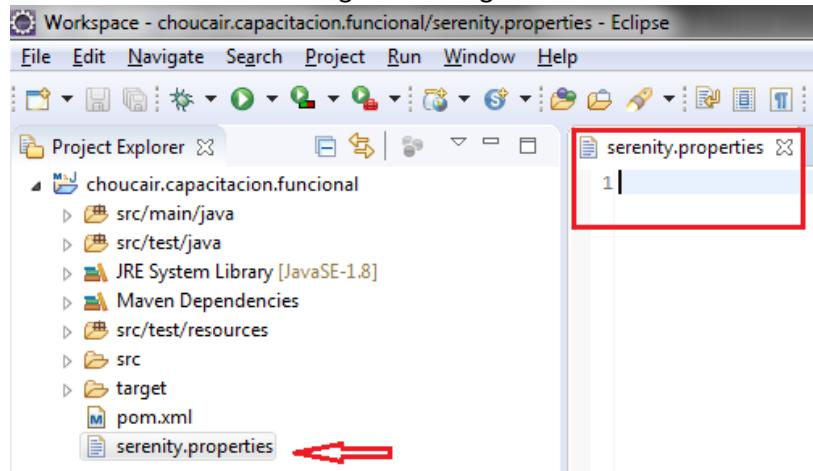
Se crea el archivo que llevará las propiedades de Serenity:



En File name: “**serenity.properties**”



Se debe visualizar como se muestra en la siguiente imagen:



INFORMACIÓN IMPORTANTE

Las propiedades y su descripción se pueden apreciar en la siguiente URL:

<https://github.com/serenity-bdd/serenity-documentation/blob/master/src/asciidoc/system-props.adoc>

CONTINÚA EJERCICIO:

Propiedades del archivo serenity.properties usadas en el framework actual:

Seleccionar las líneas color **café** copiar y pegar en la hoja **serenity.properties**

```
webdriver.driver=chrome
webdriver.chrome.driver=src/test/resources/drivers/chromedriver.exe
webdriver.firefox.driver=src/test/resources/drivers/geckodriver.exe
webdriver.ie.driver=src/test/resources/drivers/IEDriverServer.exe
serenity.project.name = Framework - Aplicativo Orquídea

serenity.use.unique.browser = false

serenity.dry.run=false

chrome.switches=--start-maximized
serenity.verbose.steps=FALSE
serenity.report.encoding=UTF8
feature.file.encoding =UTF8

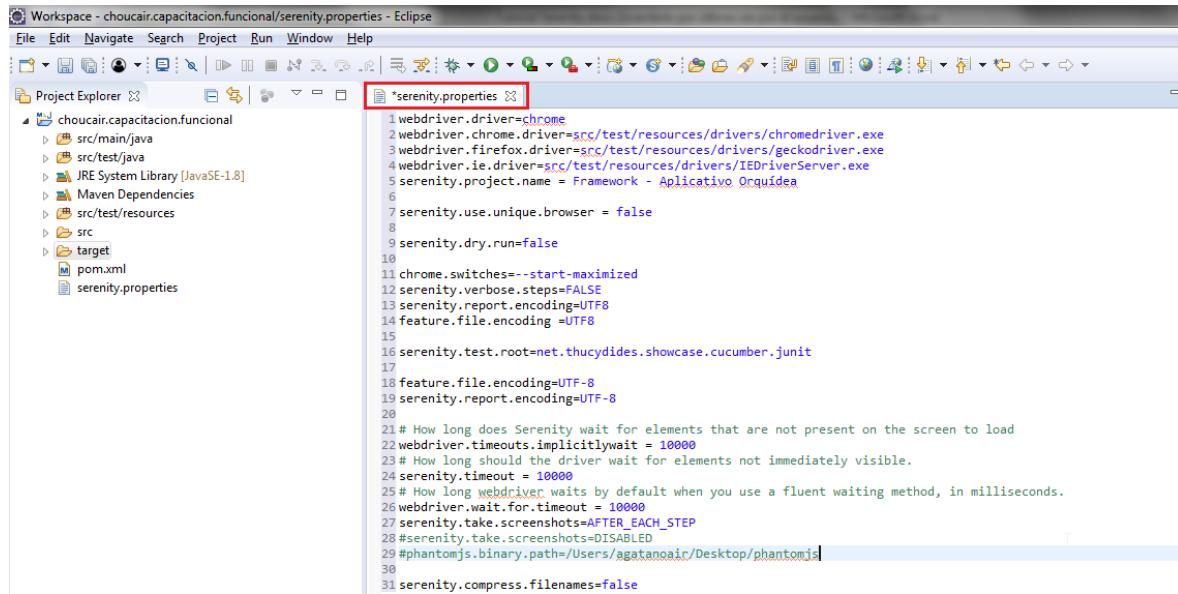
serenity.test.root=net.thucydides.showcase.cucumber.junit

feature.file.encoding=UTF-8
serenity.report.encoding=UTF-8

# How long does Serenity wait for elements that are not present on the screen to load
webdriver.timeouts.implicitlywait = 10000
# How long should the driver wait for elements not immediately visible.
serenity.timeout = 10000
# How long webdriver waits by default when you use a fluent waiting method, in milliseconds.
webdriver.wait.for.timeout = 10000
serenity.take.screenshots=AFTER_EACH_STEP
#serenity.take.screenshots=DISABLED
```

```
#phantomjs.binary.path=/Users/agatanoair/Desktop/phantomjs
serenity.compress.filenames=false
```

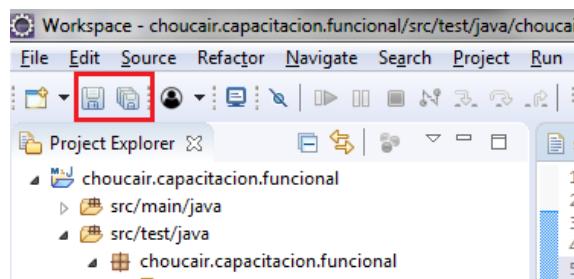
Debe quedar como indica la siguiente imagen:



```
Workspace - choucair.capacitacion.funcional/serenity.properties - Eclipse
File Edit Navigate Search Project Run Window Help
Project Explorer *serenity.properties
choucair.capacitacion.funcional
src/main/java
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src/test/resources
src
target
pom.xml
serenity.properties

1 webdriver.driver=chrome
2 webdriver.chrome.driver=src/test/resources/drivers/chromedriver.exe
3 webdriver.firefox.driver=src/test/resources/drivers/geckodriver.exe
4 webdriver.ie.driver=src/test/resources/drivers/IEDriverServer.exe
5 serenity.project.name = Framework - Aplicativo Orquídea
6
7 serenity.use.unique.browser = false
8
9 serenity.dry.run=false
10
11 chrome.switches=-start-maximized
12 serenity.verbose.steps=FALSE
13 serenity.report.encoding=UTF8
14 feature.file.encoding =UTF8
15
16 serenity.test.root=net.thucydides.showcase.cucumber.junit
17
18 feature.file.encoding=UTF-8
19 serenity.report.encoding=UTF-8
20
21# How long does Serenity wait for elements that are not present on the screen to load
22webdriver.timeouts.implicitlywait = 10000
23# How long should the driver wait for elements not immediately visible.
24serenity.timeout = 10000
25# How long webdriver waits by default when you use a fluent waiting method, in milliseconds.
26webdriver.wait.for.timeout = 10000
27serenity.take.screenshots=AFTER_EACH_STEP
28serenity.take.screenshots=DISABLED
29#phantomjs.binary.path=/Users/agatanoair/Desktop/phantomjs
30
31serenity.compress.filenames=false
```

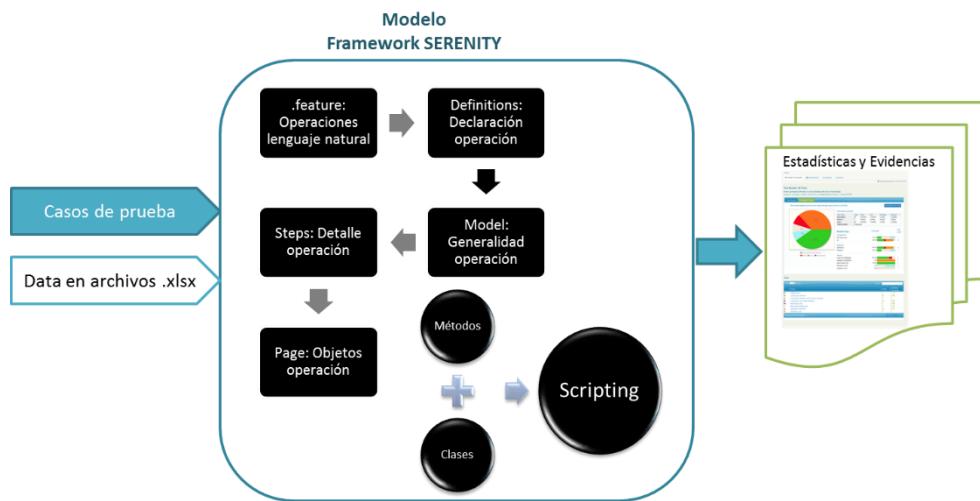
Dar Click en botón GUARDAR



INFORMACIÓN IMPORTANTE

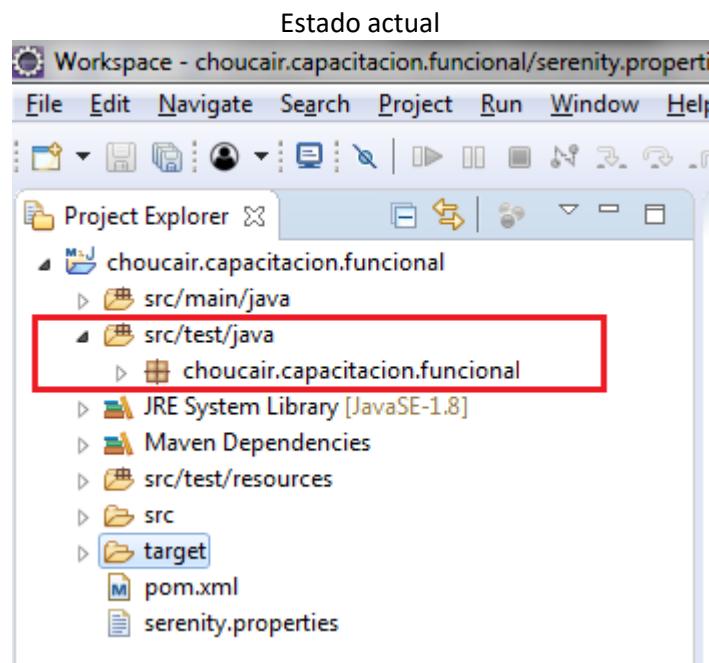
Construcción Estructura proyecto

Una vez el proyecto esté configurado con las herramientas necesarias, se procede a estructurar el framework:

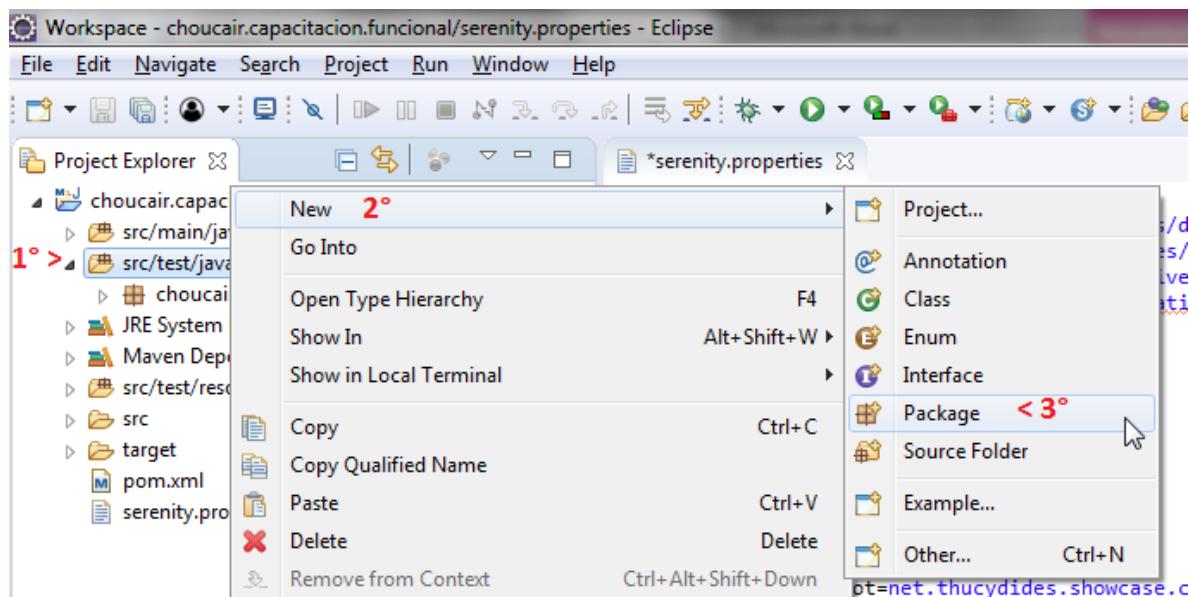


CREACIÓN DE PAQUETES

Se crean los paquetes correspondientes en el proyecto:



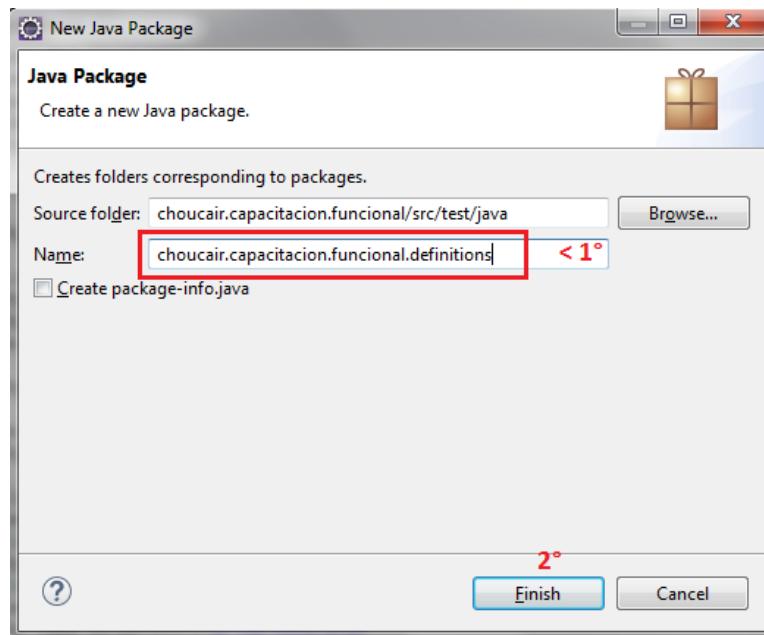
Click derecho sobre “src/test/java” para crear un paquete como indica la siguiente imagen:



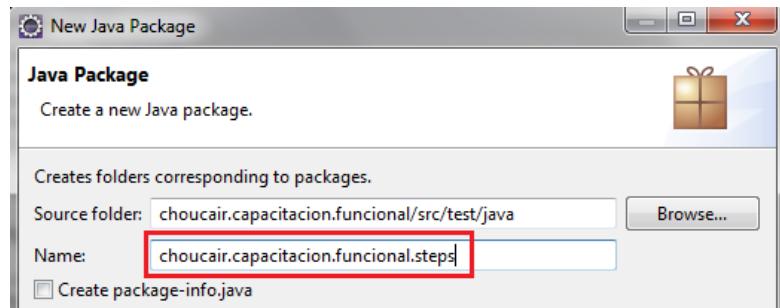
Este paso se repite para crear cada uno de los paquete se muestran a continuación

PAQUETE **DEFINITIONS**

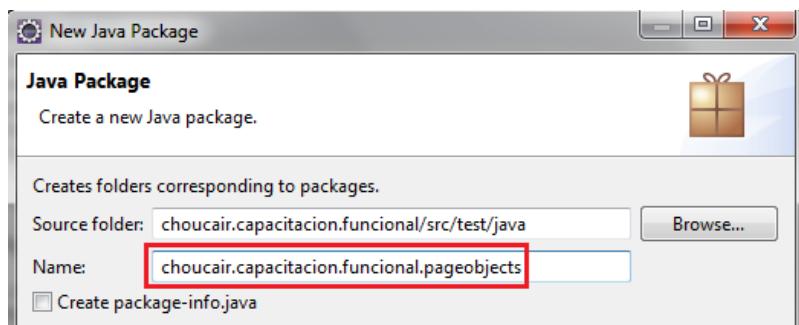
En el campo **name** se coloca el nombre del paquete que es igual al nombre del proyecto terminando en **.definitions**, ejemplo: **choucair.capacitacion.funcionaldefinitions**



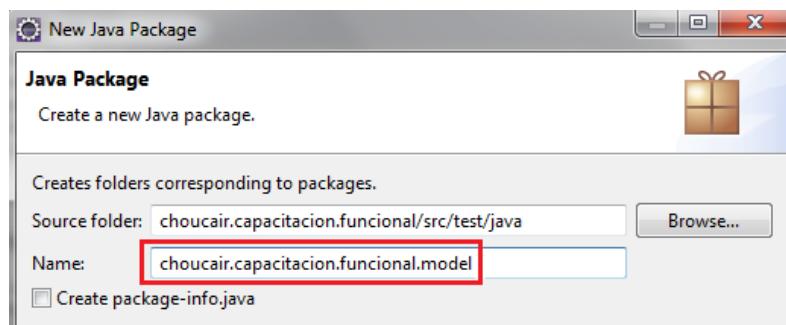
PAQUETE STEPS



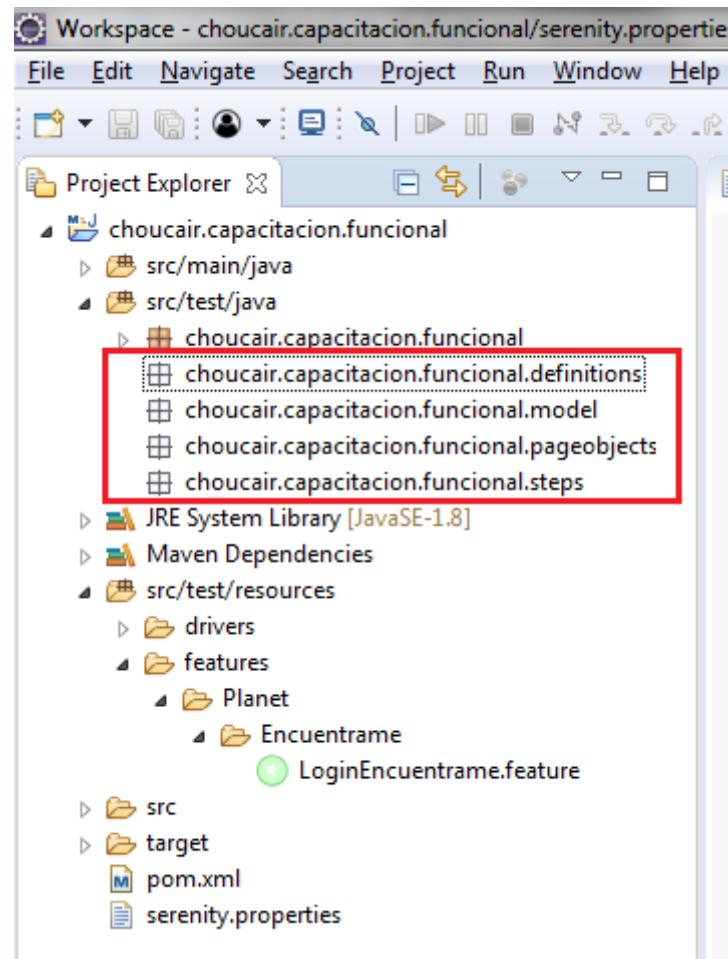
PAQUETE PAGEOBJECTS



PAQUETE MODEL



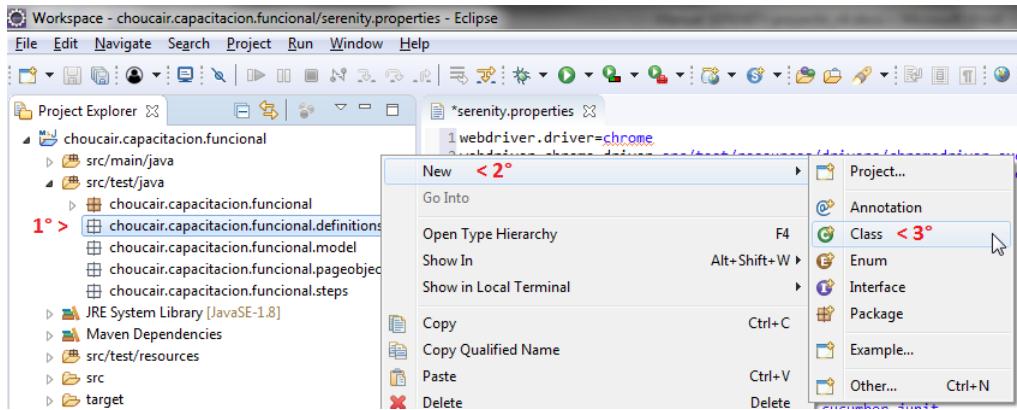
En este momento se debe visualizar los paquetes creado como indica la siguiente imagen:



CREACIÓN DE CLASES

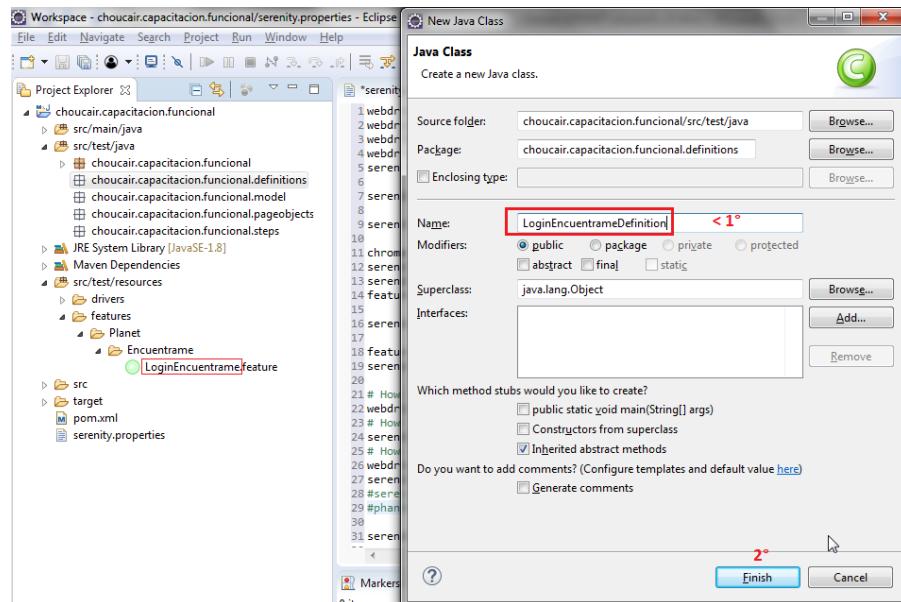
Desde éste punto se inicia con la elaboración de las diversas clases de java que soportarán la operación y los escenarios en los **features** tal como se mostró en la imagen anterior: “**Modelo Framework SERENITY**”.

Para crear la clase “**definition**”, Click derecho en el Package **choucair.capacitacion.funcional.definition**

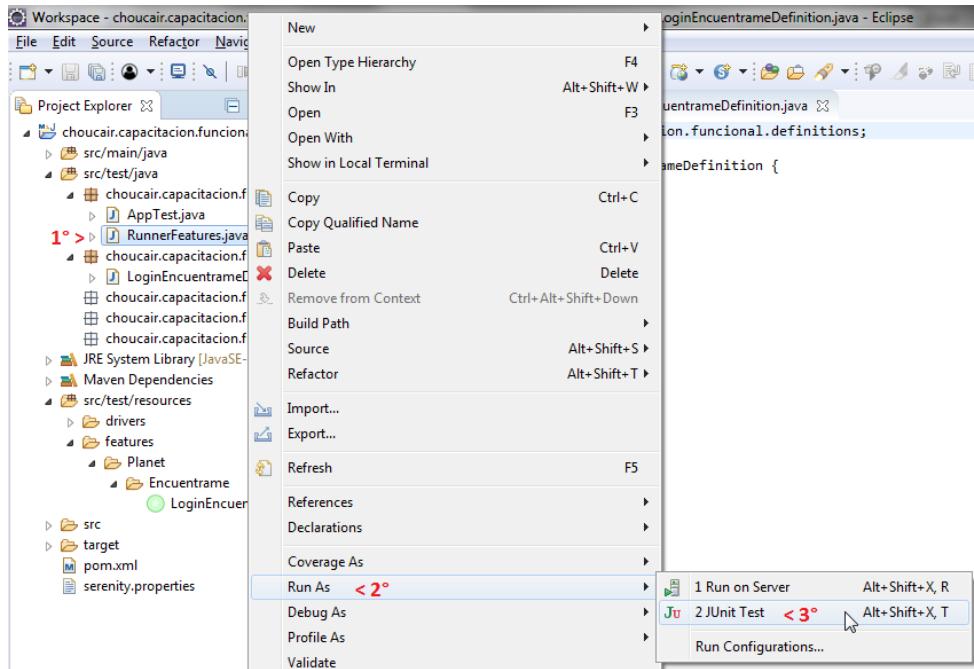


*Todas las clases creadas en este paquete tendrán el sufijo **Definition***

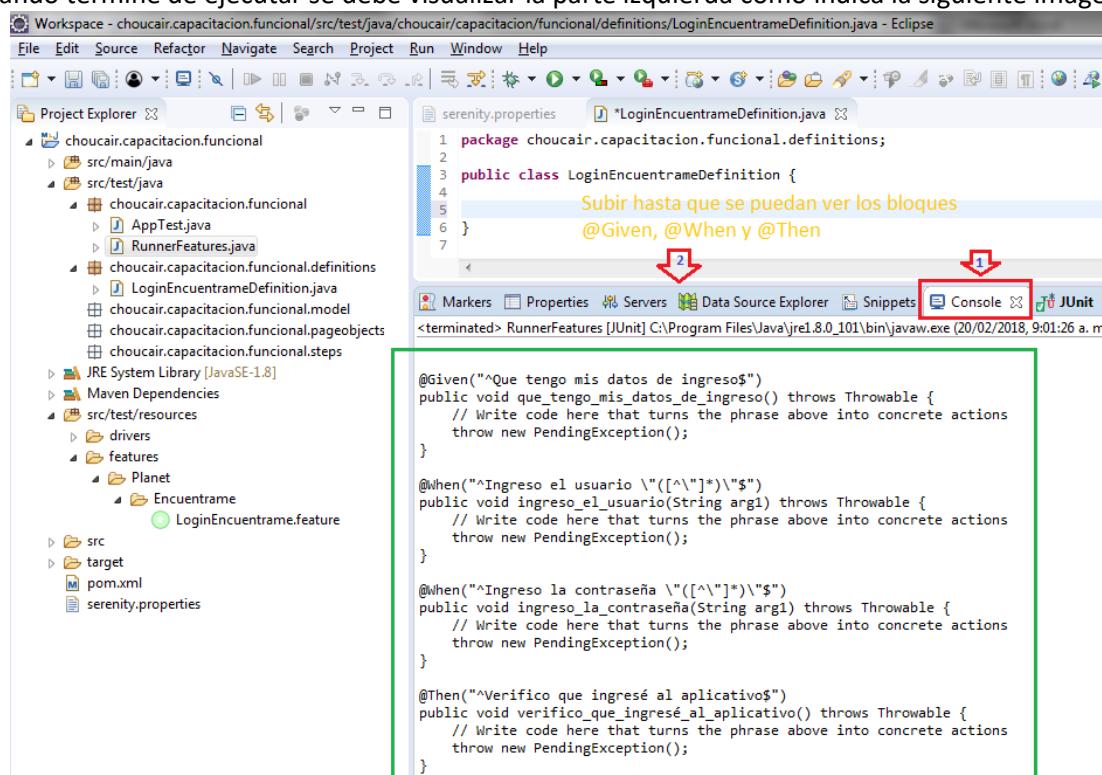
El nombre de la clase debe ser igual al nombre del .feature + Definition como se muestra en la imagen.



Se verifica que no exista error ejecutando desde la clase de java creada con antelación con JUnit, haciendo click derecho en la clase RunnerFeatures.java como indica la siguiente imagen:



Cuando termine de ejecutar se debe visualizar la parte izquierda como indica la siguiente imagen:



Seleccionar y Copiar todos los segmentos Given, When y Then: Como se indica la imagen

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/LoginPlanetDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java
1 package choucair.capacitacion.funcionaldefinitions;
2
3 public class LoginPlanetDefinition {
4
5 }
6

Markers Properties Servers Data Source Explorer Snippets Console <terminated> RunnerFeatures [JUnit] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (20/02/2018, 9:01:22)

@Given("Que tengo mis datos de ingreso$")
public void que_tengo_mis_datos_de_ingreso() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@When("^Ingreso el usuario \"([^\"]*)\"$")
public void ingreso_el_usuario(String arg1) throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@When("^Ingreso la contraseña \"([^\"]*)\"$")
public void ingreso_la_contraseña(String arg1) throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Then("^Verifico que ingresé al aplicativo$")
public void verifico_que_ingresé_al_aplicativo() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

```

Pegar en la clase creada “**LoginPlanetDefinition**”

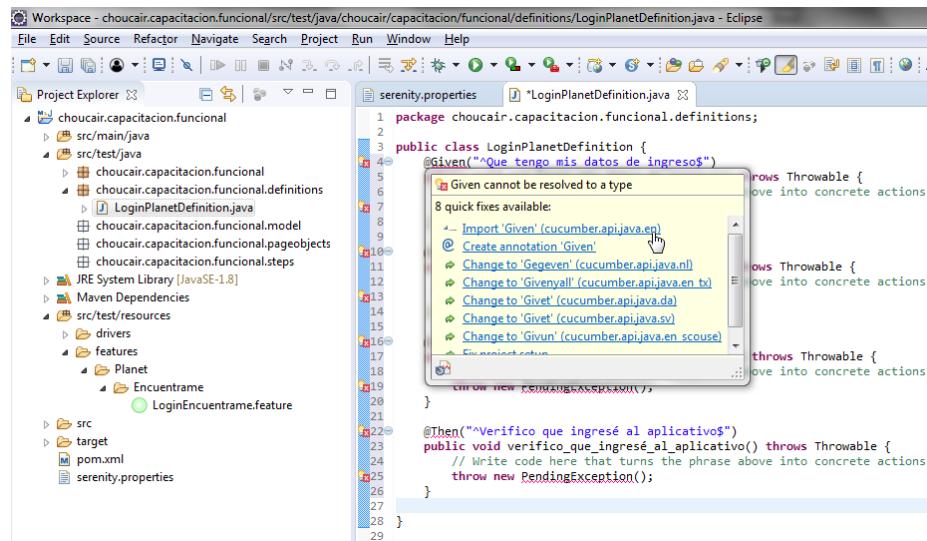
La clase de verse como muestra la siguiente imagen con varios errores:

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/LoginPlanetDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties *LoginPlanetDefinition.java
1 package choucair.capacitacion.funcionaldefinitions;
2
3 public class LoginPlanetDefinition {
4
5     @Given("Que tengo mis datos de ingresos")
6     public void que_tengo_mis_datos_de_ingreso() throws Throwable {
7         // Write code here that turns the phrase above into concrete actions
8         throw new PendingException();
9     }
10
11     @When("^Ingreso el usuario \"([^\"]*)\"$")
12     public void ingreso_el_usuario(String arg1) throws Throwable {
13         // Write code here that turns the phrase above into concrete actions
14         throw new PendingException();
15     }
16
17     @When("^Ingreso la contraseña \"([^\"]*)\"$")
18     public void ingreso_la_contraseña(String arg1) throws Throwable {
19         // Write code here that turns the phrase above into concrete actions
20         throw new PendingException();
21     }
22
23     @Then("^Verifico que ingresé al aplicativo$")
24     public void verifico_que_ingresé_al_aplicativo() throws Throwable {
25         // Write code here that turns the phrase above into concrete actions
26         throw new PendingException();
27     }
28 }

```

Se posiciona sobre las palabras Given, When y Then subrayadas con rojo y seleccionar la línea import como se indica en la imagen Import 'Given' (cucumber.api.java.en)



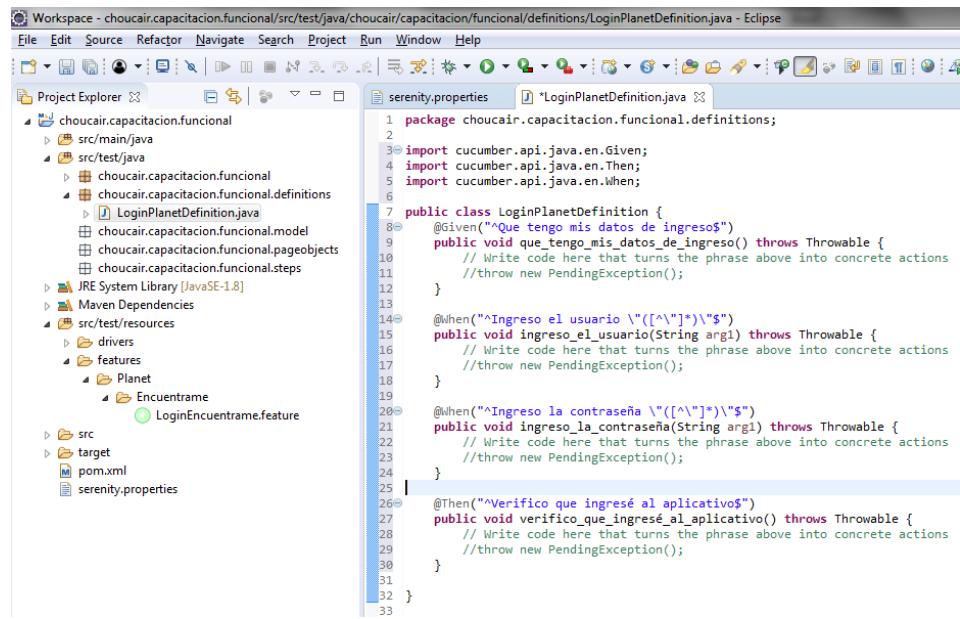
Después de corregidos los errores con el paso anterior, se debe visualizar como se indica la siguiente imagen:

```

1 package choucair.capacitacion.funcionaldefinitions;
2
3 import cucumber.api.java.en.Given;
4 import cucumber.api.java.en.Then;
5 import cucumber.api.java.en.When;
6
7 public class LoginPlanetDefinition {
8     @Given("Que tengo mis datos de ingreso$")
9     public void que_tengo_mis_datos_de_ingreso() throws Throwable {
10         // Write code here that turns the phrase above into concrete actions
11         throw new PendingException();
12     }
13
14     @When("^Ingreso el usuario \"([^\"]*)\"$")
15     public void ingreso_el_usuario(String arg1) throws Throwable {
16         // Write code here that turns the phrase above into concrete actions
17         throw new PendingException();
18     }
19
20     @When("^Ingreso la contraseña \"([^\"]*)\"$")
21     public void ingreso_la_contraseña(String arg1) throws Throwable {
22         // Write code here that turns the phrase above into concrete actions
23         throw new PendingException();
24     }
25
26     @Then("^Verifico que ingresé al aplicativo$")
27     public void verifico_que_ingresé_al_aplicativo() throws Throwable {
28         // Write code here that turns the phrase above into concrete actions
29         throw new PendingException();
30     }
31
32 }
```

En todas las líneas `throw new PendingException();` que están subrayadas con rojo se les debe anteponer `//throw new PendingException();`

Después de hacer los pasos anteriores, la clase se debe visualizar como indica la imagen, SIN errores:



```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/LoginPlanetDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties *LoginPlanetDefinition.java
1 package choucair.capacitacion.funcionaldefinitions;
2
3 import cucumber.api.java.en.Given;
4 import cucumber.api.java.en.Then;
5 import cucumber.api.java.en.When;
6
7 public class LoginPlanetDefinition {
8     @Given("^Que tengo mis datos de ingreso$")
9     public void que_tengo_mis_datos_de_ingreso() throws Throwable {
10         // Write code here that turns the phrase above into concrete actions
11         //throw new PendingException();
12     }
13
14     @When("^Ingreso el usuario \"([^\"]*)\"$")
15     public void ingreso_el_usuario(String arg1) throws Throwable {
16         // Write code here that turns the phrase above into concrete actions
17         //throw new PendingException();
18     }
19
20     @When("^Ingreso la contraseña \"([^\"]*)\"$")
21     public void ingreso_la_contraseña(String arg1) throws Throwable {
22         // Write code here that turns the phrase above into concrete actions
23         //throw new PendingException();
24     }
25
26     @Then("^Verifico que ingresé al aplicativo$")
27     public void verifico_que_ingresé_al_aplicativo() throws Throwable {
28         // Write code here that turns the phrase above into concrete actions
29         //throw new PendingException();
30     }
31
32 }
33

```

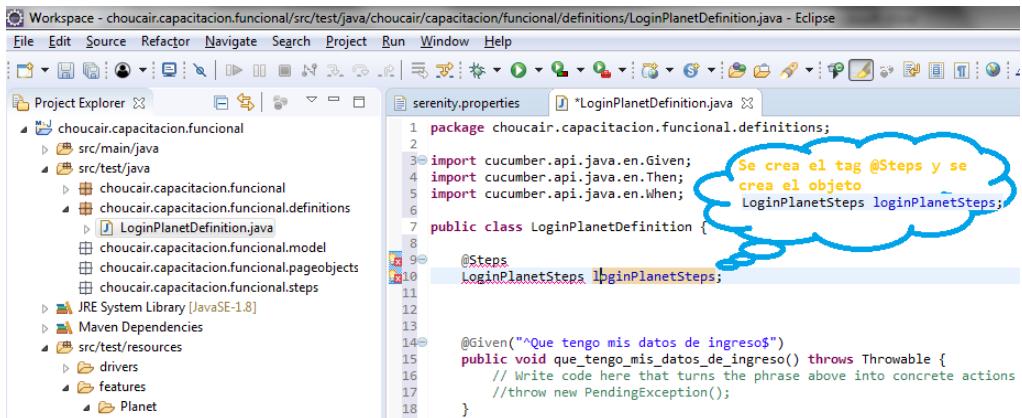
En la clase Definitions

Se agregan las líneas como indica la imagen.

```

@Steps
IngresoLoginPlanetSteps ingresoLoginPlanetSteps;

```

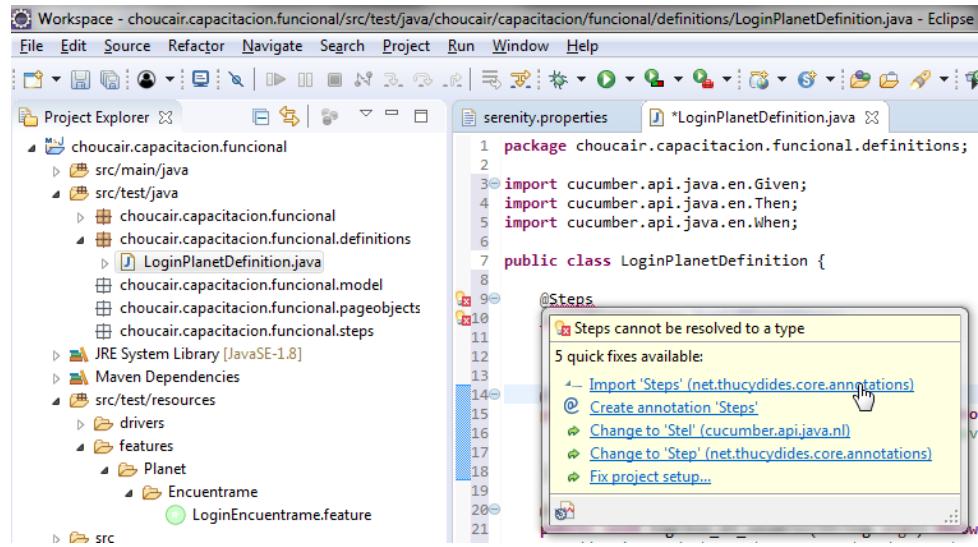


```

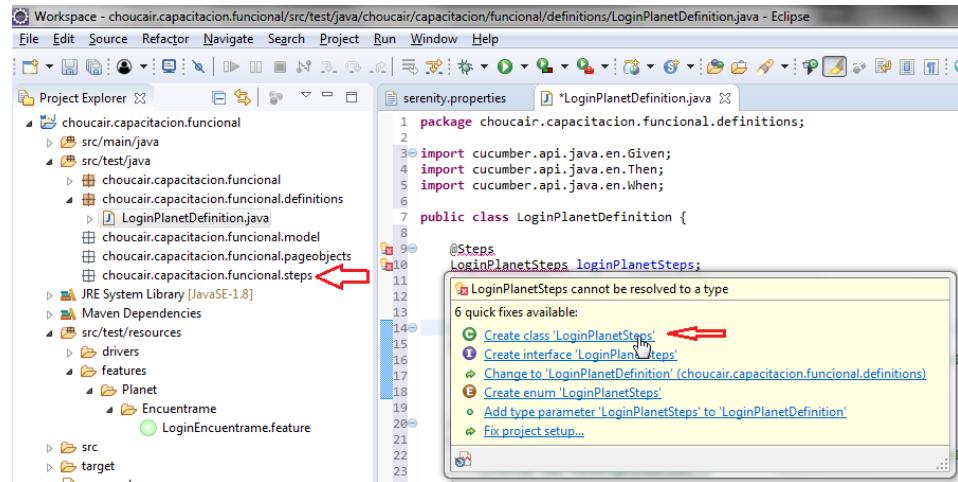
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/LoginPlanetDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties *LoginPlanetDefinition.java
1 package choucair.capacitacion.funcionaldefinitions;
2
3 import cucumber.api.java.en.Given;
4 import cucumber.api.java.en.Then;
5 import cucumber.api.java.en.When;
6
7 public class LoginPlanetDefinition {
8
9     @Steps
10     LoginPlanetSteps loginPlanetSteps;
11
12
13
14     @Given("^Que tengo mis datos de ingreso$")
15     public void que_tengo_mis_datos_de_ingreso() throws Throwable {
16         // Write code here that turns the phrase above into concrete actions
17         //throw new PendingException();
18     }
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

```

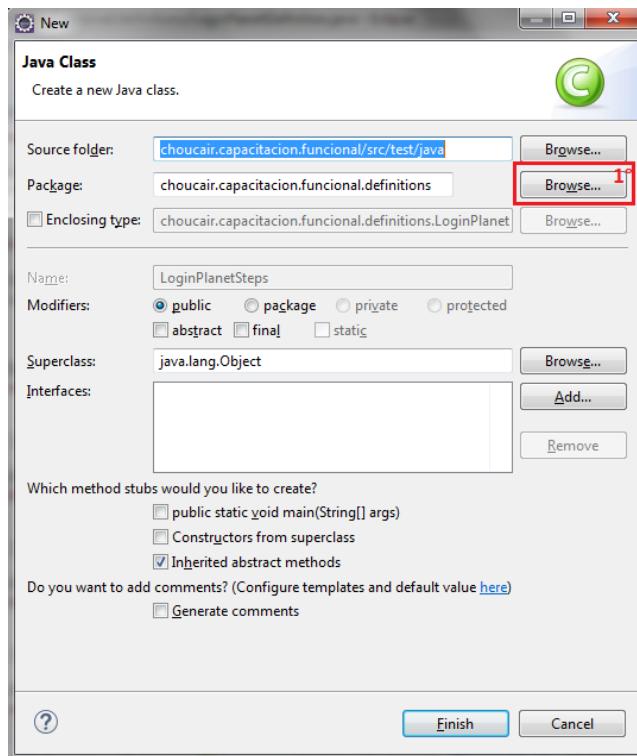
Para quitar el primer error: ubicar el cursor sobre la palabra `@Steps` y en la ventana emergente seleccionar como indica la imagen [Import 'Steps' \(net.thucydides.core.annotations\)](#):



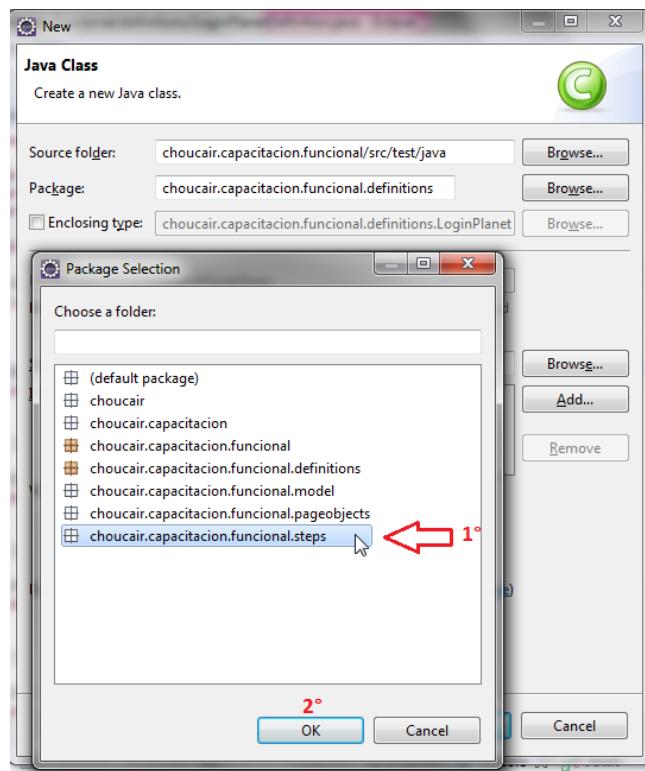
Para quitar el segundo error: ubicar el cursor sobre la palabra `LoginPlanetSteps` y en la ventana emergente seleccionar como indica la imagen [Create class 'xxxxxxxxxx'](#):



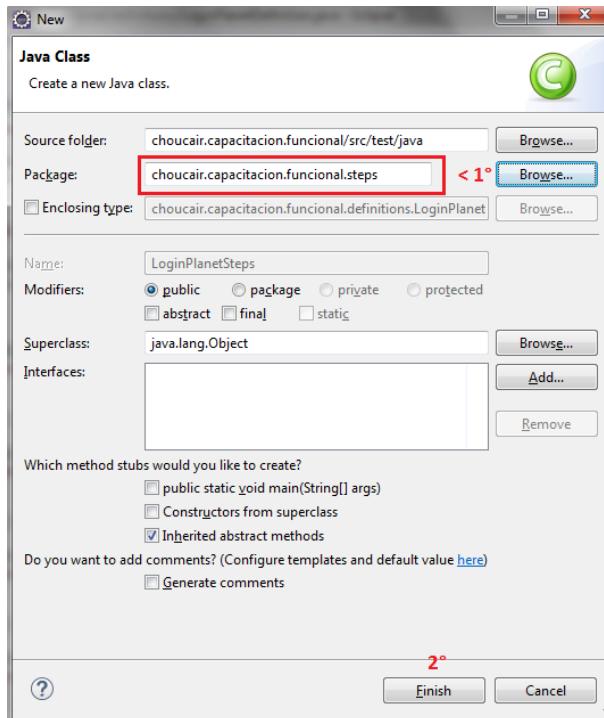
Después de hacer la selección en la ventana emergente se abrirá la siguiente pantalla, la cual pide que se indique en cual paquete debe ser creada la clase, se da click en **Browse...**:



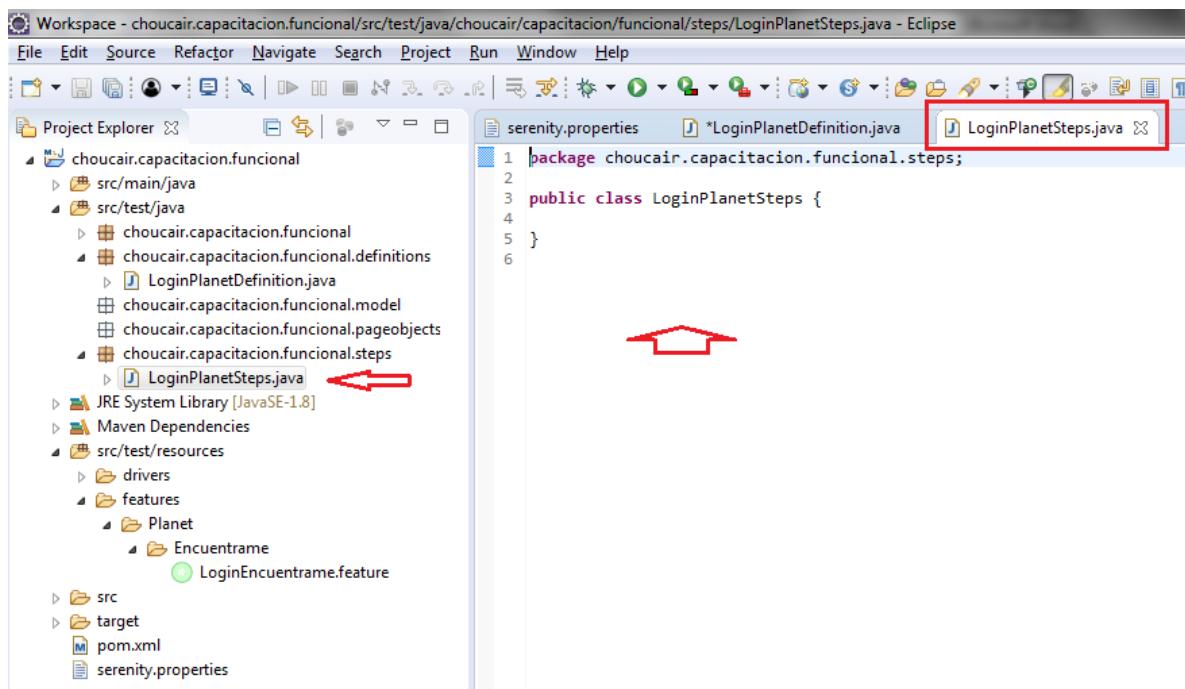
Seleccionar paquete donde debe ser creada la clase



Después de seleccionado el paquete la ventana en el campo **Package:** se verá el nombre del paquete seleccionado como indica la siguiente imagen para después dar click a **Finish**:



La clase **LoginPlanetSteps** creada quedará en el paquete “**choucair.capacitacion.funcional.steps**” como se ve en la siguiente imagen:



La clase “LoginPlanetDefinition” quedaría sin errores como se muestra en la siguiente imagen:

```
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/LoginPlanetDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties *LoginPlanetDefinition.java LoginPlanetSteps.java
choucair.capacitacion.funcional
src/main/java
src/test/java
choucair.capacitacion.funcional
choucair.capacitacion.funcional.definitions
choucair.capacitacion.funcional.model
choucair.capacitacion.funcional.pageobjects
choucair.capacitacion.funcional.steps
JRE System Library [JavaSE-1.8]
Maven Dependencies
src/test/resources
drivers
features
Planet
Encuentrame
LoginEncuentrame.feature
src
target
pom.xml
serenity.properties
3@ import choucair.capacitacion.funcional.steps.LoginPlanetSteps;
4 import cucumber.api.java.en.Given;
5 import cucumber.api.java.en.Then;
6 import cucumber.api.java.en.When;
7 import net.thucydides.core.annotations.Steps;
8
9 public class LoginPlanetDefinition {
10     @Steps
11     LoginPlanetSteps loginPlanetSteps;
12
13
14
15
16@ Given("^Que tengo mis datos de ingreso$")
17 public void que_tengo_mis_datos_de_ingreso() throws Throwable {
18     // Write code here that turns the phrase above into concrete actions
19     //throw new PendingException();
20 }
21
22@ When("^Ingreso el usuario \"([^\"]*)\"$")
23 public void ingreso_el_usuario(String arg1) throws Throwable {
24     // Write code here that turns the phrase above into concrete actions
25     //throw new PendingException();
26 }
27
28@ When("^Ingreso la contraseña \"([^\"]*)\"$")
29 public void ingreso_la_contraseña(String arg1) throws Throwable {
30     // Write code here that turns the phrase above into concrete actions
31     //throw new PendingException();
32 }
33
34@ Then("^Verifico que ingresé al aplicativo$")
35 public void verifico_que_ingresé_al_aplicativo() throws Throwable {
```

En la clase **LoginPlanetDefinition**:

los argumentos que vienen por defecto `arg1` (`String arg1`) deben ser cambiados según lo definido en el archivo `.feature`

```
9 public class LoginPlanetDefinition {
10
11    @Steps
12    LoginPlanetSteps loginPlanetSteps;
13
14
15
16@ Given("^Que tengo mis datos de ingreso$")
17 public void que_tengo_mis_datos_de_ingreso() throws Throwable {
18     // Write code here that turns the phrase above into concrete actions
19     //throw new PendingException();
20 }
21
22@ When("^Ingreso el usuario \"([^\"]*)\"$")
23 public void ingreso_el_usuario(String arg1) throws Throwable {
24     // Write code here that turns the phrase above into concrete actions
25     //throw new PendingException();
26 }
27
28@ When("^Ingreso la contraseña \"([^\"]*)\"$")
29 public void ingreso_la_contraseña(String arg1) throws Throwable {
30     // Write code here that turns the phrase above into concrete actions
31     //throw new PendingException();
32 }
33
34@ Then("^Verifico que ingresé al aplicativo$")
35 public void verifico_que_ingresé_al_aplicativo() throws Throwable {
```

Desde la clase **IngresoLoginPlanetDefinitions** se hace llamado a los métodos de la clase **LoginPlanetSteps** creada anteriormente, como se puede ver en la imagen, eclipse nos indica que hay errores en las llamadas por que los métodos NO están creados, tener en cuenta que el llamado se hace enviando el parámetro recibido como indican las marcas en verde:

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/LoginPlanetDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java LoginPlanetSteps.java
choucair.capacitacion.funcional
src/main/java
src/test/java
choucair.capacitacion.funcional
choucair.capacitacion.funcionaldefinitions
LoginPlanetDefinition.java
choucair.capacitacion.funcional.model
choucair.capacitacion.funcional.pageobjects
choucair.capacitacion.funcional.steps
LoginPlanetSteps.java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src/test/resources
drivers
features
Planet
Encuentrame
LoginEncuentrame.feature
src
target
pom.xml
serenity.properties

import net.thucydides.core.annotations.Steps;
public class LoginPlanetDefinition {
    @Steps
    LoginPlanetSteps loginPlanetSteps;

    @Given("^Que tengo mis datos de ingreso$")
    public void que_tengo_mis_datos_de_ingreso() throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.AbrirUrlPlanet(); <-- Red arrow
        //throw new PendingException();
    }

    @When("^Ingreso el usuario \"([^\"]*)\"$")
    public void ingreso_el_usuario(String usuario) throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.Ingresouuario(usuario); <-- Red arrow
        //throw new PendingException();
    }

    @When("^Ingreso la contraseña \"([^\"]*)\"$")
    public void ingreso_la_contraseña(String clave) throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.Ingresoclave(clave); <-- Red arrow
        //throw new PendingException();
    }

    @Then("^Verifico que ingresé al aplicativo$")
    public void verifico_que_ingresé_al_aplicativo() throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.Verificaringreso(); <-- Red arrow
        //throw new PendingException();
    }
}

```

Para solucionar los errores mostrados en con flechas rojas en la imagen anterior se hace lo siguiente:

Colocar el cursor encima del error (palabra subrayada con rojo) y seleccionar la opción: ([Create method 'XXXXXX\(\)' in type 'LoginPlanetSteps'](#)) según muestra la imagen siguiente:

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/LoginPlanetDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java LoginPlanetSteps.java
choucair.capacitacion.funcional
src/main/java
src/test/java
choucair.capacitacion.funcional
choucair.capacitacion.funcionaldefinitions
LoginPlanetDefinition.java
choucair.capacitacion.funcional.model
choucair.capacitacion.funcional.pageobjects
choucair.capacitacion.funcional.steps
LoginPlanetSteps.java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src/test/resources
drivers
features
Planet
Encuentrame
LoginEncuentrame.feature
src
target
pom.xml
serenity.properties

import net.thucydides.core.annotations.Steps;
public class LoginPlanetDefinition {
    @Steps
    LoginPlanetSteps loginPlanetSteps;

    @Given("^Que tengo mis datos de ingreso$")
    public void que_tengo_mis_datos_de_ingreso() throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.AbrirUrlPlanet(); <-- Red arrow
        //throw new PendingException();
    }

    @When("^Ingreso el usuario \"([^\"]*)\"$")
    public void ingreso_el_usuario(String usuario) throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.Ingresouuario(usuario); <-- Red arrow
        //throw new PendingException();
    }

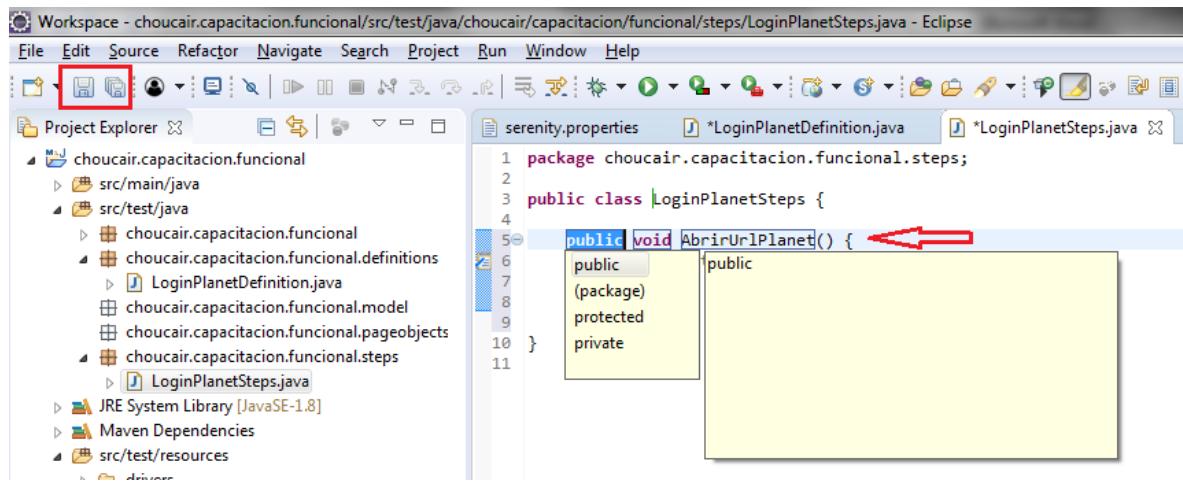
    @When("^Ingreso la contraseña \"([^\"]*)\"$")
    public void ingreso_la_contraseña(String clave) throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.Ingresoclave(clave); <-- Red arrow
        //throw new PendingException();
    }

    @Then("^Verifico que ingresé al aplicativo$")
    public void verifico_que_ingresé_al_aplicativo() throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.Verificaringreso(); <-- Red arrow
        //throw new PendingException();
    }
}

2 quick fixes available:
Create method 'AbrirUrlPlanet()' in type 'LoginPlanetSteps' <-- Red arrow
Add cast to 'loginPlanetSteps'
Press F2 for focus

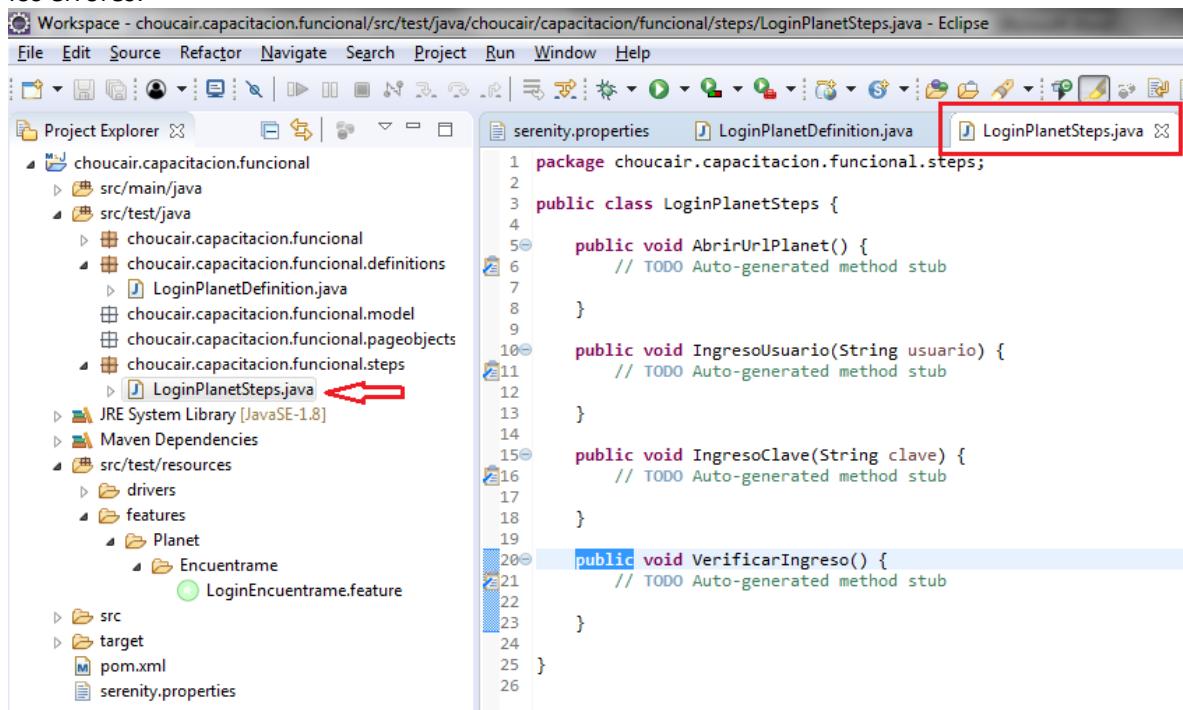
```

El método es creado en la clase y se da click en el botón guardar

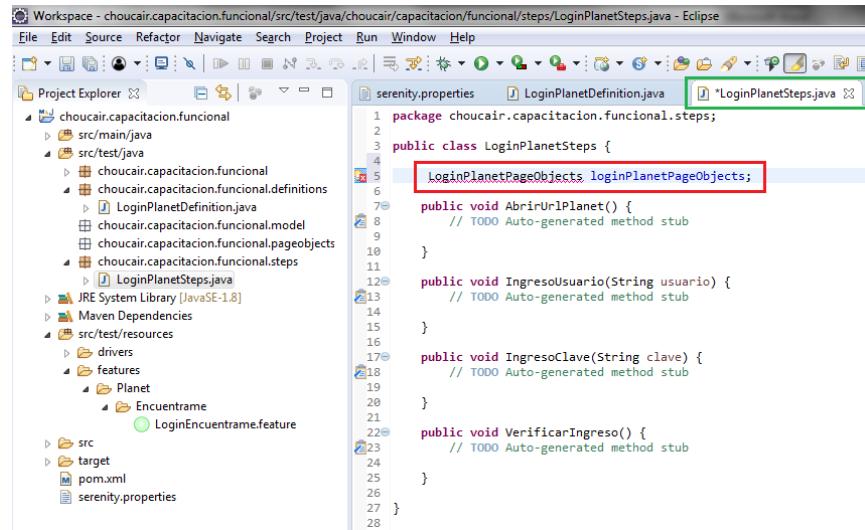


Este mismo procedimiento se hace con cada uno de los errores de este tipo que aparezcan, luego se da click en el botón guardar.

La clase Steps debe verse como indica la imagen después de hacer el paso anterior en cada uno de los errores:



Dentro de la clase “Steps” (LoginPlanetSteps) se hace la declaración del objeto **pageobject** como se indica en la imagen, también se pueden apreciar los métodos creados anteriormente:



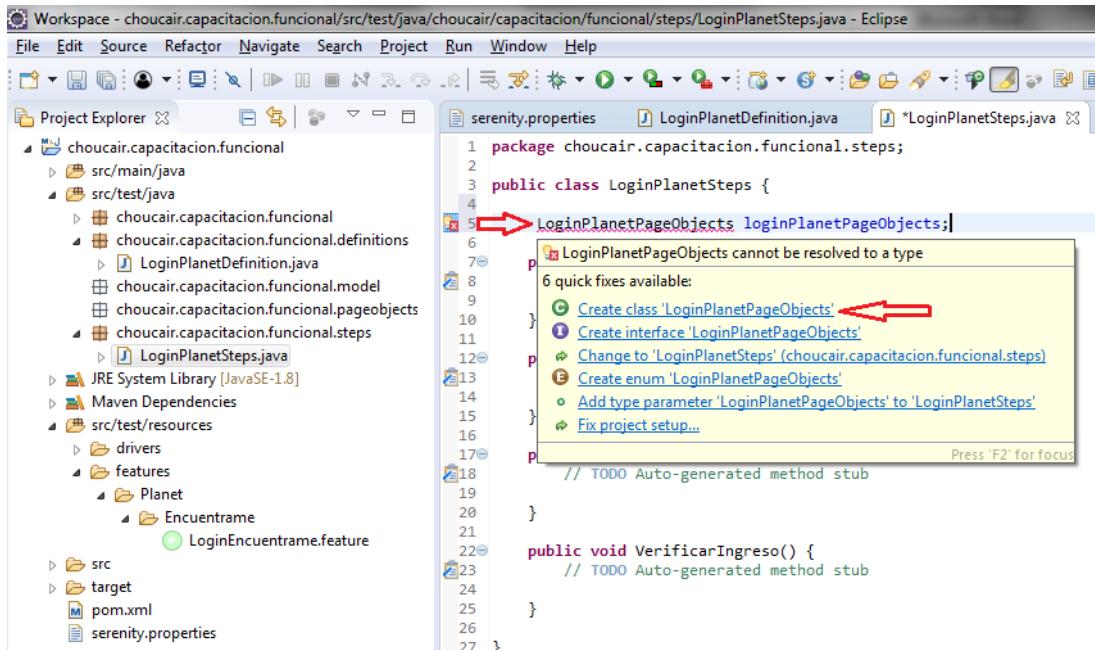
```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/steps/LoginPlanetSteps.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java *LoginPlanetSteps.java
1 package choucair.capacitacion.funcional.steps;
2
3 public class LoginPlanetSteps {
4     LoginPlanetPageObjects loginPlanetPageObjects;
5
6     public void AbrirUrlPlanet() {
7         // TODO Auto-generated method stub
8     }
9
10    public void IngresoUsuario(String usuario) {
11        // TODO Auto-generated method stub
12    }
13    public void Ingresoclave(String clave) {
14        // TODO Auto-generated method stub
15    }
16    public void VerificarIngreso() {
17        // TODO Auto-generated method stub
18    }
19
20}
21
22}
23
24
25
26
27
28

```

La imagen muestra que eclipse reporta un error, porque la clase NO está construida.

Se procede de la misma manera que los errores anteriores, se posiciona el cursor encima del error y se selecciona [Create class ‘XxxxxxXXXX’](#):

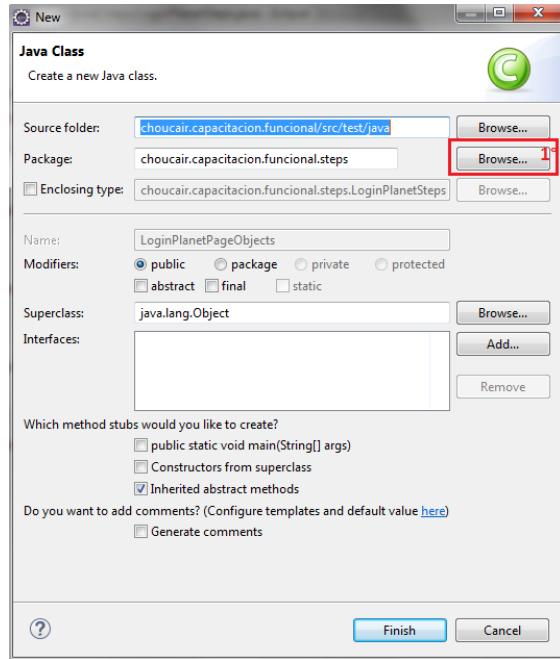


```

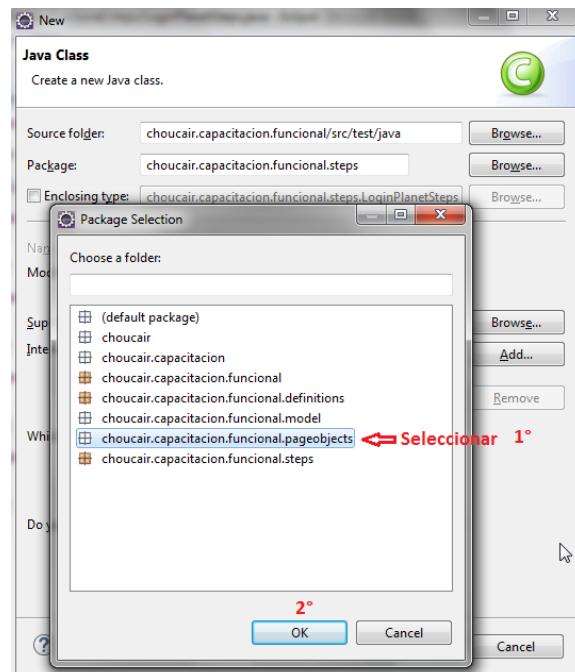
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/steps/LoginPlanetSteps.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java *LoginPlanetSteps.java
1 package choucair.capacitacion.funcional.steps;
2
3 public class LoginPlanetSteps {
4     LoginPlanetPageObjects loginPlanetPageObjects;
5
6     public void AbrirUrlPlanet() {
7         // TODO Auto-generated method stub
8     }
9
10    public void IngresoUsuario(String usuario) {
11        // TODO Auto-generated method stub
12    }
13    public void Ingresoclave(String clave) {
14        // TODO Auto-generated method stub
15    }
16    public void VerificarIngreso() {
17        // TODO Auto-generated method stub
18    }
19
20}
21
22}
23
24
25
26
27
28

```

En la ventana que aparece hacer click en **Browse...** para poder seleccionar el paquete donde debe ser creada la clase (**choucair.capacitacion.funcional.pageobjects**):

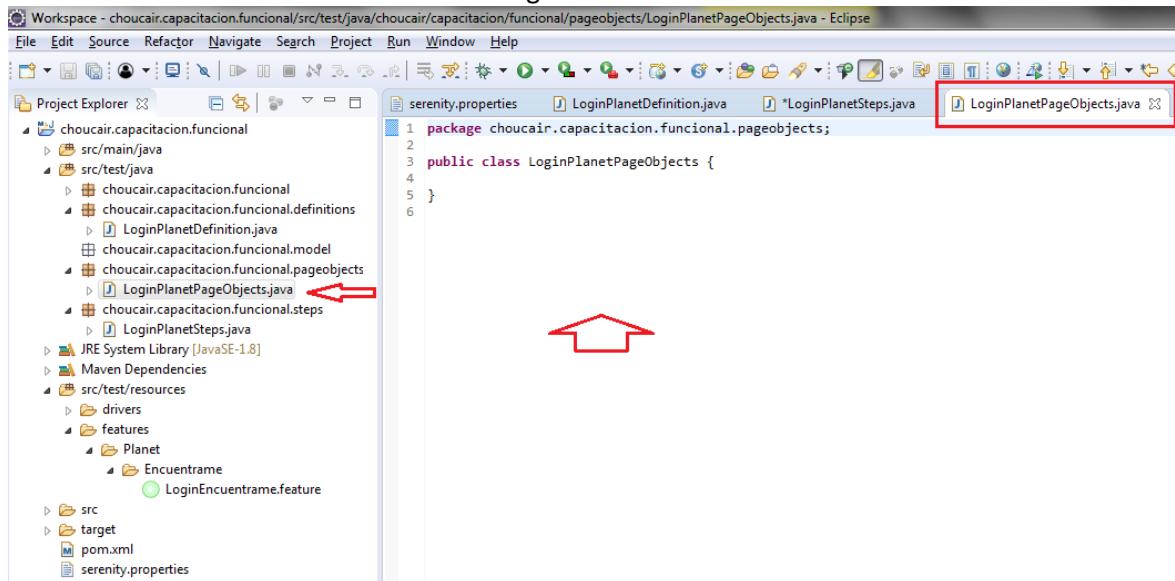


En la ventana “**Package Selection**” seleccionar el paquete “**pageobjects**” y dar click en **OK**



En la ventana que queda abierta se debe apreciar el paquete seleccionado “choucair.capacitacion.funcional.pageobjects”

La clase es creada se debe visualizar de la siguiente manera:

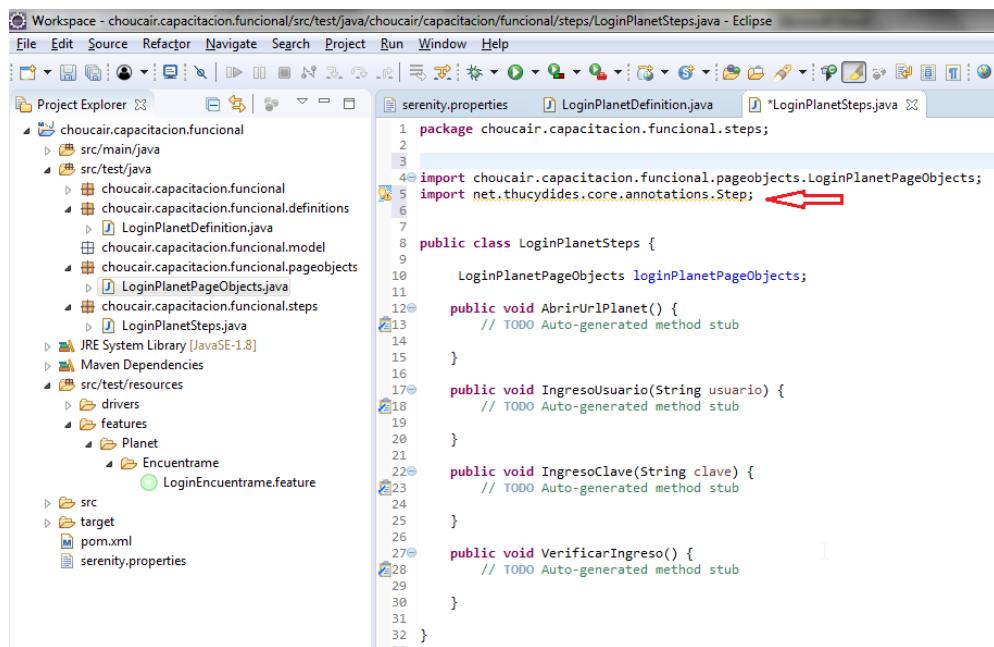


```
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/pageobjects/LoginPlanetPageObjects.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java *LoginPlanetSteps.java LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.pageobjects;
2
3 public class LoginPlanetPageObjects {
4
5 }
```

Se ingresan los siguientes **import** en la clase Steps `LoginPlanetSteps`:

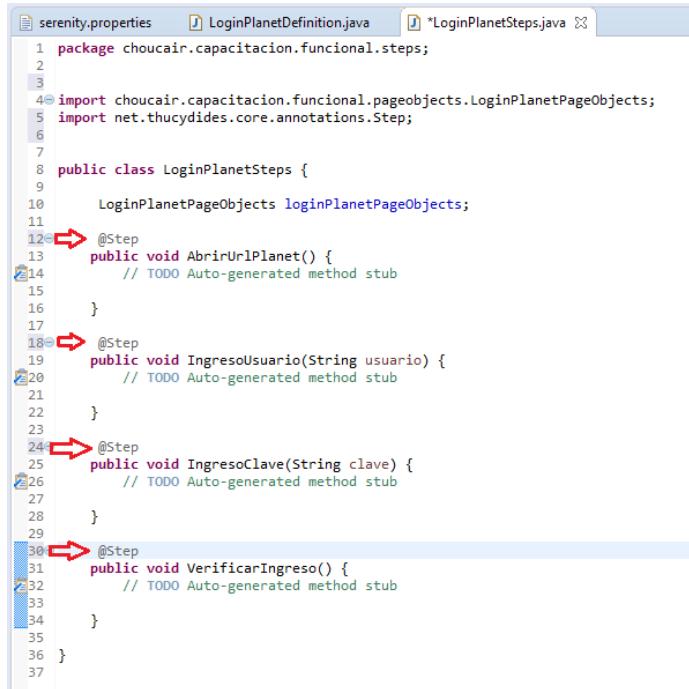
```
import net.thucydides.core.annotations.Step;
```

Como indica la siguiente imagen:



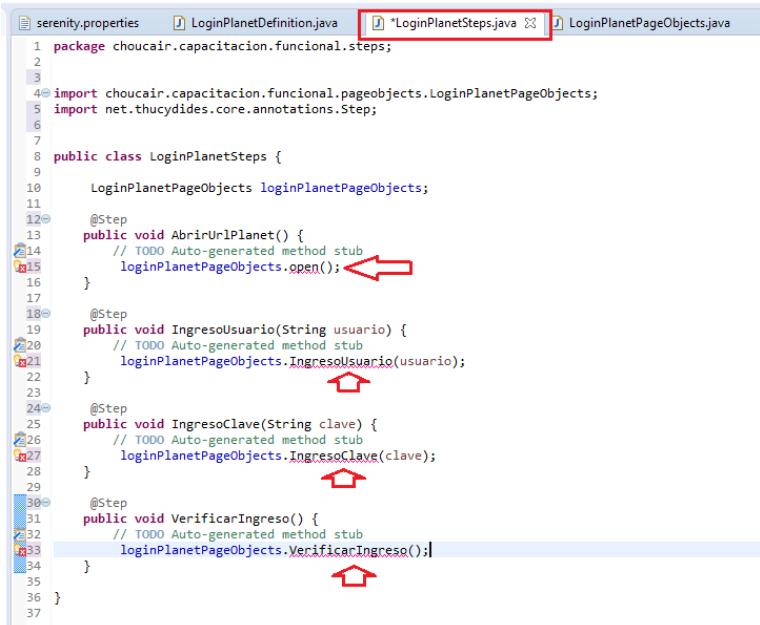
```
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/steps/LoginPlanetSteps.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java *LoginPlanetSteps.java
1 package choucair.capacitacion.funcional.steps;
2
3
4 import choucair.capacitacion.funcional.pageobjects.LoginPlanetPageObjects;
5 import net.thucydides.core.annotations.Step; ←
6
7
8 public class LoginPlanetSteps {
9
10     LoginPlanetPageObjects loginPlanetPageObjects;
11
12     public void AbrirUrlPlanet() {
13         // TODO Auto-generated method stub
14     }
15
16     public void IngresoUsuario(String usuario) {
17         // TODO Auto-generated method stub
18     }
19
20     public void IngresoClave(String clave) {
21         // TODO Auto-generated method stub
22     }
23
24     public void VerificarIngreso() {
25         // TODO Auto-generated method stub
26     }
27
28 }
```

Antes de cada método se escribe el tag @Step como se ve en la siguiente imagen



```
serenity.properties LoginPlanetDefinition.java *LoginPlanetSteps.java LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.steps;
2
3 import choucair.capacitacion.funcional.pageobjects.LoginPlanetPageObjects;
4 import net.thucydides.core.annotations.Step;
5
6
7 public class LoginPlanetSteps {
8     LoginPlanetPageObjects loginPlanetPageObjects;
9
10    @Step
11    public void AbrirUrlPlanet() {
12        // TODO Auto-generated method stub
13    }
14
15    @Step
16    public void IngresoUsuario(String usuario) {
17        // TODO Auto-generated method stub
18    }
19
20    @Step
21    public void IngresoClave(String clave) {
22        // TODO Auto-generated method stub
23    }
24
25    @Step
26    public void VerificarIngreso() {
27        // TODO Auto-generated method stub
28    }
29
30}
31
32
33
34
35
36
37
```

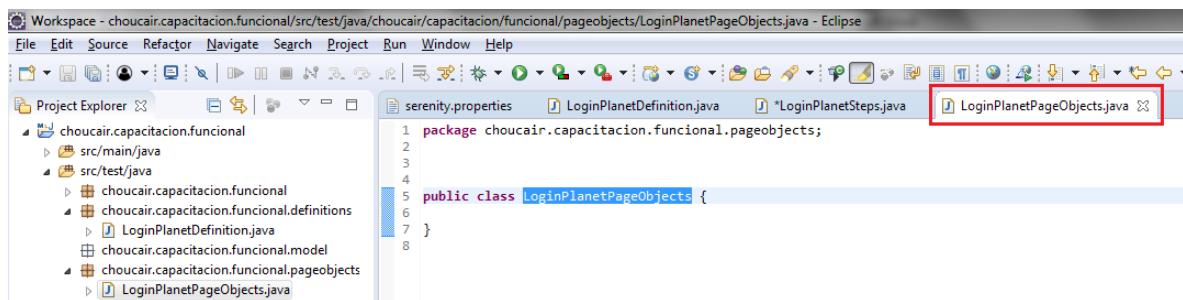
Se hacen los llamados a los métodos de la clase **pageobjects** (`LoginPlanetPageObjects`) como indica la imagen, se evidencia que eclipse reporta errores porque los métodos NO están creados en la clase **pageobjects**:



```
serenity.properties LoginPlanetDefinition.java *LoginPlanetSteps.java LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.steps;
2
3 import choucair.capacitacion.funcional.pageobjects.LoginPlanetPageObjects;
4 import net.thucydides.core.annotations.Step;
5
6
7 public class LoginPlanetSteps {
8     LoginPlanetPageObjects loginPlanetPageObjects;
9
10    @Step
11    public void AbrirUrlPlanet() {
12        // TODO Auto-generated method stub
13        loginPlanetPageObjects.open(); <-- Red arrow
14    }
15
16    @Step
17    public void IngresoUsuario(String usuario) {
18        // TODO Auto-generated method stub
19        loginPlanetPageObjects.IngresoUsuario(usuario); <-- Red arrow
20    }
21
22    @Step
23    public void Ingresoclave(String clave) {
24        // TODO Auto-generated method stub
25        loginPlanetPageObjects.Ingresoclave(clave); <-- Red arrow
26    }
27
28    @Step
29    public void VerificarIngreso() {
30        // TODO Auto-generated method stub
31        loginPlanetPageObjects.VerificarIngreso(); <-- Red arrow
32    }
33
34
35
36
37}
```

Para solucionar el primer error:

El de la línea donde se llama al método “open()” `loginPlanetPageObjects.open();` se debe abrir la clase `LoginPlanetPageObjects`:

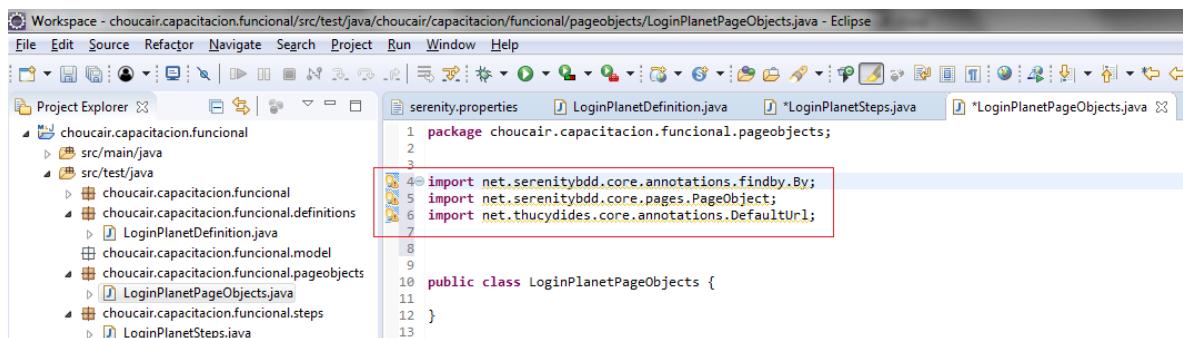


The screenshot shows the Eclipse IDE interface with the title bar "Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/pageobjects/LoginPlanetPageObjects.java - Eclipse". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The Project Explorer on the left shows the project structure under "choucair.capacitacion.funcional". The editor on the right contains the code for `LoginPlanetPageObjects.java`:

```
1 package choucair.capacitacion.funcional.pageobjects;
2
3
4 public class LoginPlanetPageObjects {
5
6 }
7
```

Se ingresan los siguientes import

```
import net.serenitybdd.core.annotations.findby.By;
import net.serenitybdd.core.pages.PageObject;
import net.thucydides.core.annotations.DefaultUrl;
```

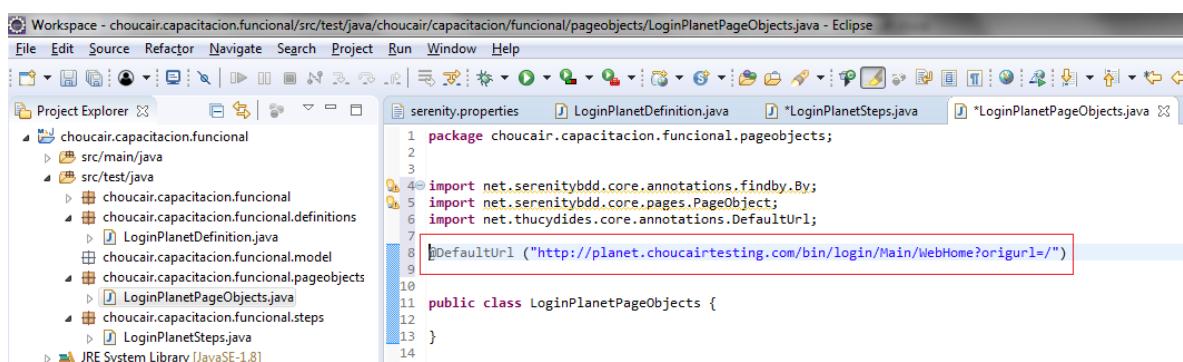


The screenshot shows the Eclipse IDE interface with the title bar "Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/pageobjects/LoginPlanetPageObjects.java - Eclipse". The editor on the right contains the code for `LoginPlanetPageObjects.java`, with three new import statements highlighted by a red box:

```
1 package choucair.capacitacion.funcional.pageobjects;
2
3
4 ④ import net.serenitybdd.core.annotations.findby.By;
5 import net.serenitybdd.core.pages.PageObject;
6 import net.thucydides.core.annotations.DefaultUrl;
7
8
9
10 public class LoginPlanetPageObjects {
11
12 }
13
```

Se agrega la siguiente línea:

```
@DefaultUrl ("http://planet.choucairtesting.com/bin/login/Main/WebHome?origurl=/")
```



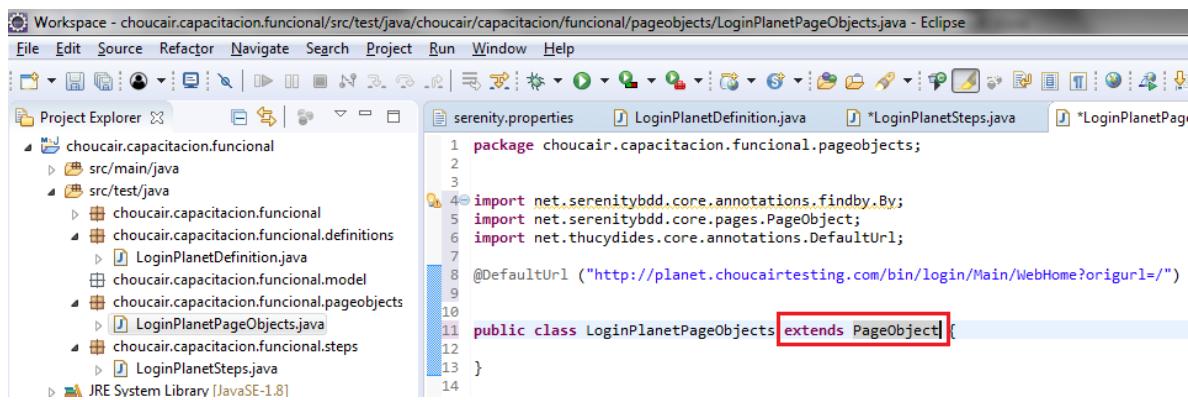
The screenshot shows the Eclipse IDE interface with the title bar "Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/pageobjects/LoginPlanetPageObjects.java - Eclipse". The editor on the right contains the code for `LoginPlanetPageObjects.java`, with the `@DefaultUrl` annotation highlighted by a red box:

```
1 package choucair.capacitacion.funcional.pageobjects;
2
3
4 ④ import net.serenitybdd.core.annotations.findby.By;
5 import net.serenitybdd.core.pages.PageObject;
6 import net.thucydides.core.annotations.DefaultUrl;
7
8 ⑤ @DefaultUrl ("http://planet.choucairtesting.com/bin/login/Main/WebHome?origurl=/")
9
10
11 public class LoginPlanetPageObjects {
12
13 }
```

Al nombre de la clase `LoginPlanetPageObjects` se le adiciona la siguiente cadena:

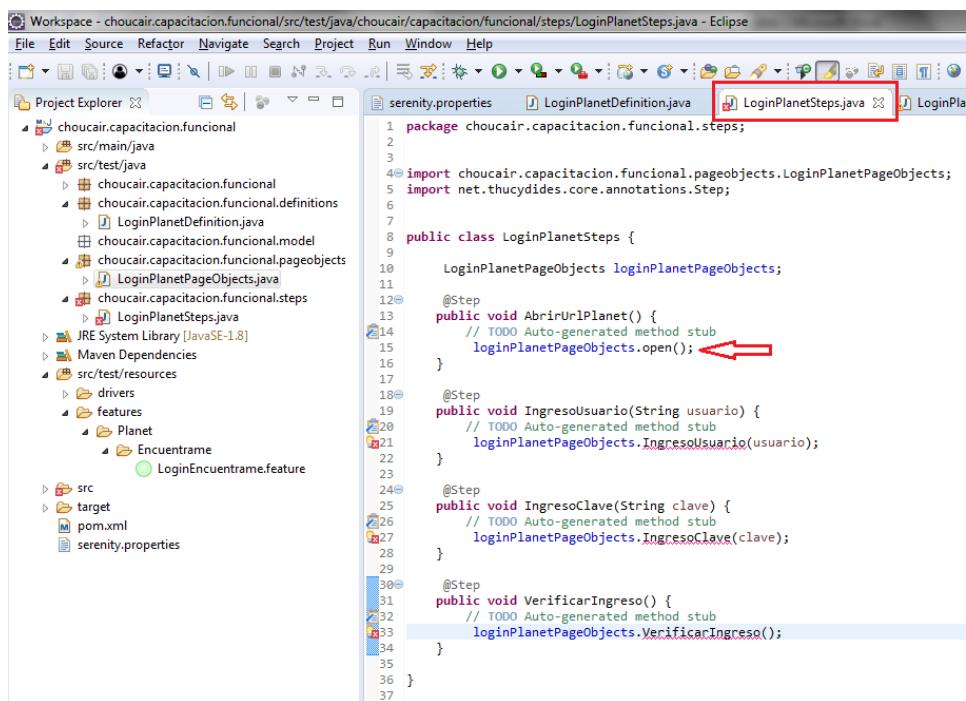
`extends PageObject`

Como se aprecia en la siguiente imagen:



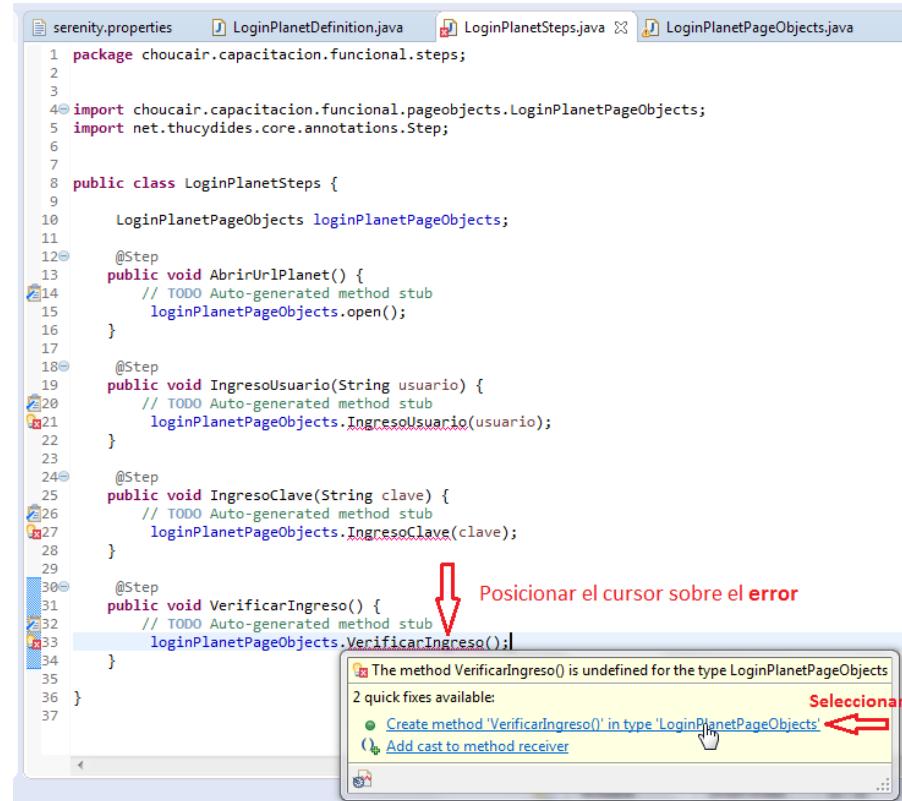
```
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/pageobjects/LoginPlanetPageObjects.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java *LoginPlanetSteps.java *LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.pageobjects;
2
3
4 import net.serenitybdd.core.annotations.findby.By;
5 import net.serenitybdd.core.pages.PageObject;
6 import net.thucydides.core.annotations.DefaultUrl;
7
8 @DefaultUrl ("http://planet.choucairtesting.com/bin/login/Main/WebHome?origurl=/")
9
10 public class LoginPlanetPageObjects extends PageObject {
11
12 }
13
14
```

Se selecciona la clase steps, se da click en botón guardar, se evidenciará que el error de la línea que indica la flecha roja ha desaparecido:



```
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/steps/LoginPlanetSteps.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java LoginPlanetSteps.java LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.steps;
2
3
4 import choucair.capacitacion.funcional.pageobjects.LoginPlanetPageObjects;
5 import net.thucydides.core.annotations.Step;
6
7
8 public class LoginPlanetSteps {
9
10     LoginPlanetPageObjects loginPlanetPageObjects;
11
12     @Step
13     public void AbrirUrlPlanet() {
14         // TODO Auto-generated method stub
15         loginPlanetPageObjects.open(); ←
16     }
17
18     @Step
19     public void IngresoUsuario(String usuario) {
20         // TODO Auto-generated method stub
21         loginPlanetPageObjects.Ingresousuario(usuario);
22     }
23
24     @Step
25     public void IngresoClave(String clave) {
26         // TODO Auto-generated method stub
27         loginPlanetPageObjects.Ingresoclave(clave);
28     }
29
30     @Step
31     public void VerificarIngreso() {
32         // TODO Auto-generated method stub
33         loginPlanetPageObjects.VerificarIngreso();
34     }
35
36 }
```

Igual que con los errores anteriores posicionar el cursor sobre el error y seleccionar como indica la imagen:



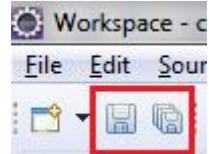
```
serenity.properties LoginPlanetDefinition.java LoginPlanetSteps.java LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.steps;
2
3
4 import choucair.capacitacion.funcional.pageobjects.LoginPlanetPageObjects;
5 import net.thucydides.core.annotations.Step;
6
7
8 public class LoginPlanetSteps {
9
10     LoginPlanetPageObjects loginPlanetPageObjects;
11
12     @Step
13     public void AbrirUrlPlanet() {
14         // TODO Auto-generated method stub
15         loginPlanetPageObjects.open();
16     }
17
18     @Step
19     public void IngresoUsuario(String usuario) {
20         // TODO Auto-generated method stub
21         loginPlanetPageObjects.IngresoUsuario(usuario);
22     }
23
24     @Step
25     public void IngresoClave(String clave) {
26         // TODO Auto-generated method stub
27         loginPlanetPageObjects.IngresoClave(clave);
28     }
29
30     @Step
31     public void VerificarIngreso() {
32         // TODO Auto-generated method stub
33         loginPlanetPageObjects.VerificarIngreso();
34     }
35
36 }
```

Posicionar el cursor sobre el error

Seleccionar

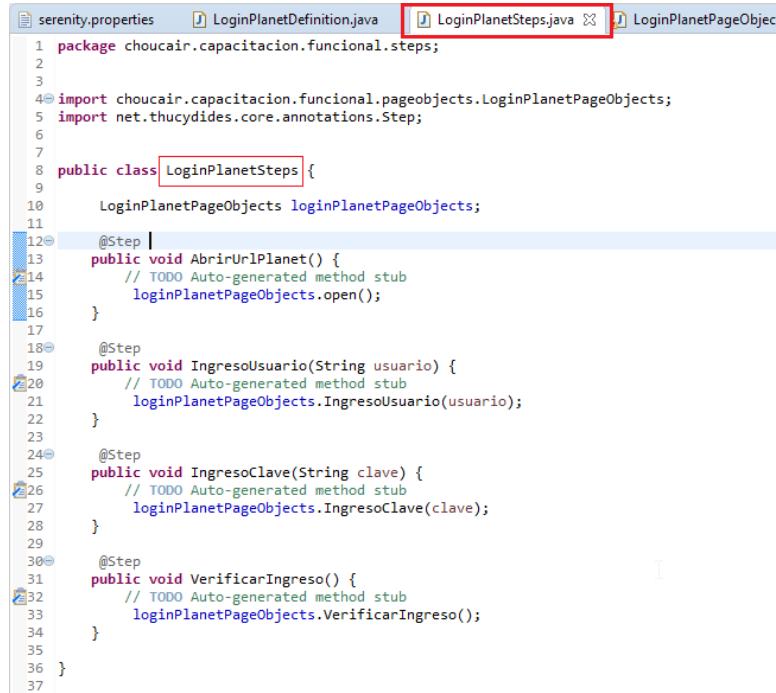
The method VerificarIngreso() is undefined for the type LoginPlanetPageObjects
2 quick fixes available:
● Create method 'VerificarIngreso()' in type 'LoginPlanetPageObjects'
● Add cast to method receiver

Dar click en el botón guardar después de cada operación en cada error.



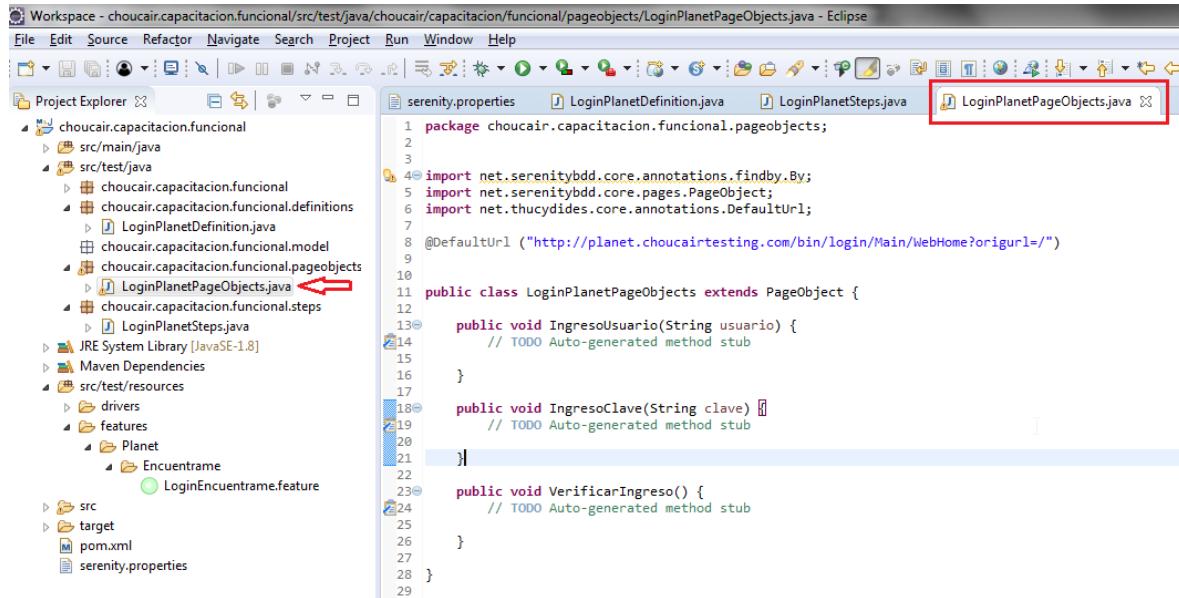
Al final la clase LoginPlanetSteps

Debe verse como indica la imagen, sin errores:



```
serenity.properties LoginPlanetDefinition.java LoginPlanetSteps.java LoginPlanetPageObject
1 package choucair.capacitacion.funcional.steps;
2
3
4 import choucair.capacitacion.funcional.pageobjects.LoginPlanetPageObjects;
5 import net.thucydides.core.annotations.Step;
6
7
8 public class LoginPlanetSteps {
9
10     LoginPlanetPageObjects loginPlanetPageObjects;
11
12     @Step
13     public void AbrirUrlPlanet() {
14         // TODO Auto-generated method stub
15         loginPlanetPageObjects.open();
16     }
17
18     @Step
19     public void IngresoUsuario(String usuario) {
20         // TODO Auto-generated method stub
21         loginPlanetPageObjects.IngresoUsuario(usuario);
22     }
23
24     @Step
25     public void IngresoClave(String clave) {
26         // TODO Auto-generated method stub
27         loginPlanetPageObjects.IngresoClave(clave);
28     }
29
30     @Step
31     public void VerificarIngreso() {
32         // TODO Auto-generated method stub
33         loginPlanetPageObjects.VerificarIngreso();
34     }
35
36 }
37
```

Y la clase LoginPlanetPageObjects debe verse como indica la siguiente imagen:



```
Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/pageobjects/LoginPlanetPageObjects.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
src/main/java
src/test/java
    choucair.capacitacion.funcional
        src/main/java
        src/test/java
            choucair.capacitacion.funcional
                src/main/java
                src/test/java
                    choucair.capacitacion.funcional
                        definitions
                            LoginPlanetDefinition.java
                        model
                        pageobjects
                            LoginPlanetPageObjects.java
                        steps
                            LoginPlanetSteps.java
                    LoginPlanetSteps.java
    Maven Dependencies
    src/test/resources
        drivers
        features
            Planet
                Encuentrame
                    LoginEncuentrame.feature
    pom.xml
    serenity.properties
serenity.properties LoginPlanetDefinition.java LoginPlanetSteps.java LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.pageobjects;
2
3
4 import net.serenitybdd.core.annotations.findby.By;
5 import net.serenitybdd.core.pages.PageObject;
6 import net.thucydides.core.annotations.DefaultUrl;
7
8 @DefaultUrl ("http://planet.choucairtesting.com/bin/login/Main/WebHome?origurl=/")
9
10 public class LoginPlanetPageObjects extends PageObject {
11
12     public void IngresoUsuario(String usuario) {
13         // TODO Auto-generated method stub
14     }
15
16     public void IngresoClave(String clave) {
17         // TODO Auto-generated method stub
18     }
19
20 }
21
22     public void VerificarIngreso() {
23         // TODO Auto-generated method stub
24     }
25
26 }
27
28 }
```

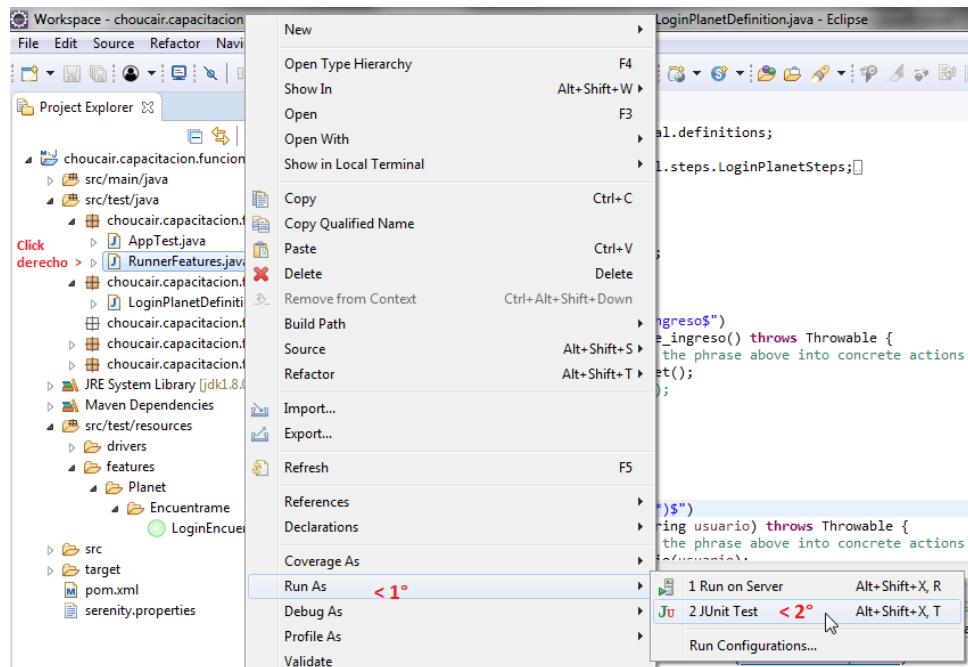
En la clase `LoginPlanetPageObjects` se hace el mapeo de los objetos y se implementa cada método como indica la imagen:

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/pageobjects/LoginPlanetPageObjects.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer serenity.properties LoginPlanetDefinition.java LoginPlanetSteps.java *LoginPlanetPageObjects.java
1 package choucair.capacitacion.funcional.pageobjects;
2
3
4 import net.serenitybdd.core.annotations.findby.By;
5 import net.serenitybdd.core.pages.PageObject;
6 import net.thucydides.core.annotations.DefaultUrl;
7
8 @DefaultUrl ("http://planet.choucairtesting.com/bin/login/Main/WebHome?origurl=/")
9
10 public class LoginPlanetPageObjects extends PageObject {
11
12     public void IngresoUsuario(String usuario) {
13         // TODO Auto-generated method stub
14         find(By.name("username")).type(usuario); Propiedad name
15     }
16
17     public void IngresoClave(String clave) {
18         // TODO Auto-generated method stub
19         find(By.name("password")).typeAndEnter(clave); Propiedad name
20     }
21
22     public void VerificarIngreso() {
23         // TODO Auto-generated method stub
24         findBy("//*[@id='patternSideBarContents']/span[1]/a").click(); Propiedad Xpath
25     }
26
27
28 }

```

EJECUCIÓN PROYECTO

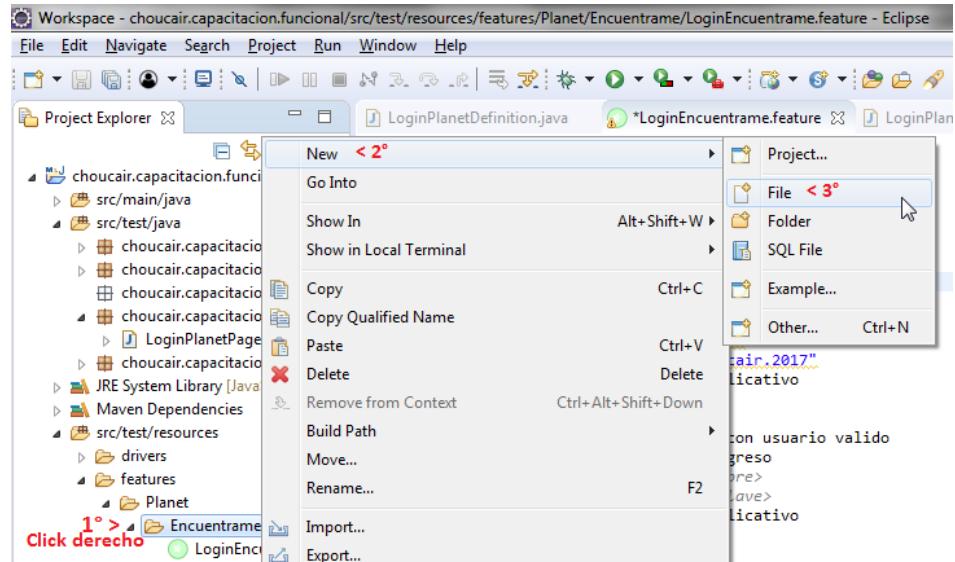


Haciendo todos los pasos del tutorial se abrirá el navegador y hará un logueo en la página de planet.

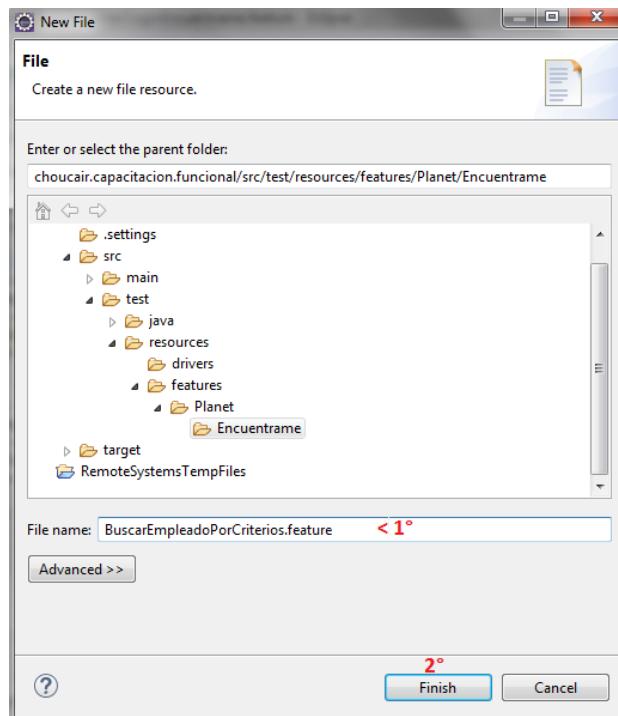
AUTOMATIZACIÓN FUNCIONALIDAD BUSQUEDA EMPLEADO (Encuéntrame)

Cada vez que se requiera automatizar una nueva funcionalidad se deben seguir los mismos pasos.

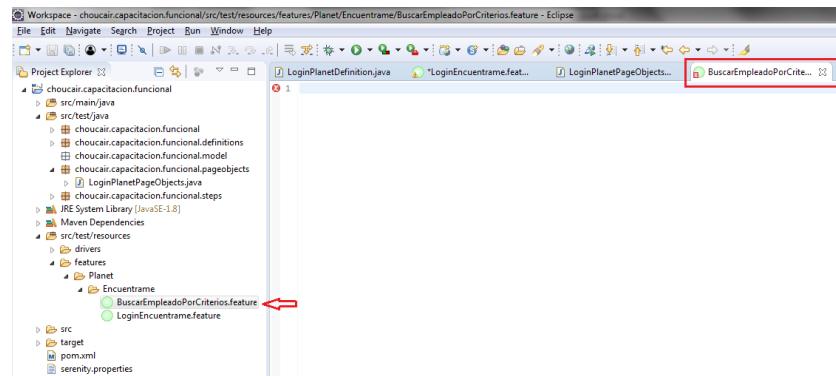
Creación del archivo .feature:



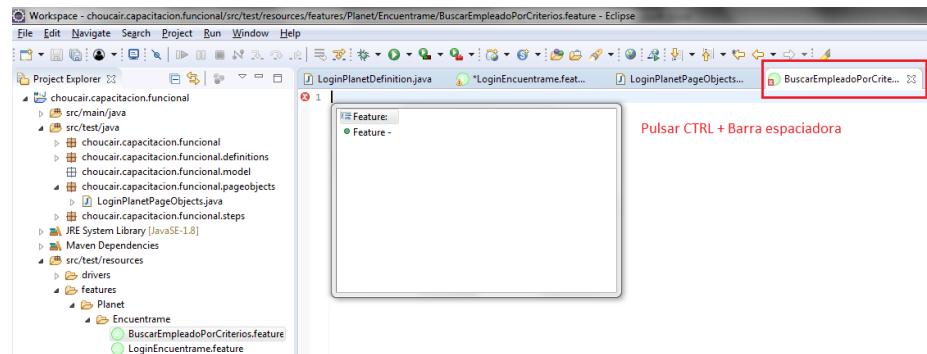
Se da el nombre al archivo .feature según la funcionalidad que se quiere automatizar:



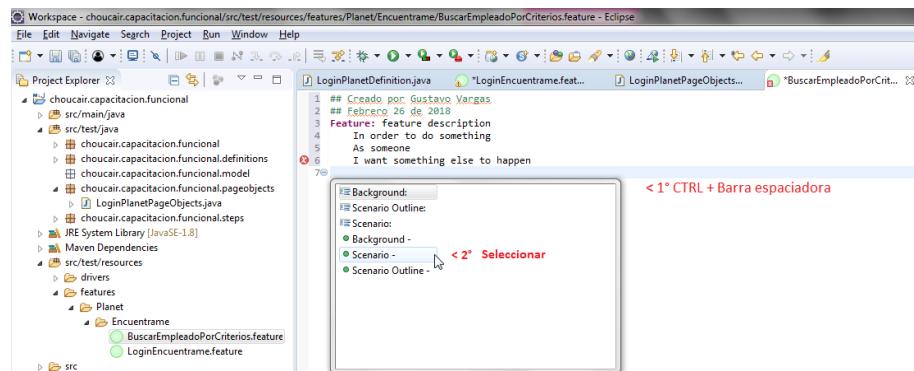
El archivo creado debe verse como indica la imagen siguiente:



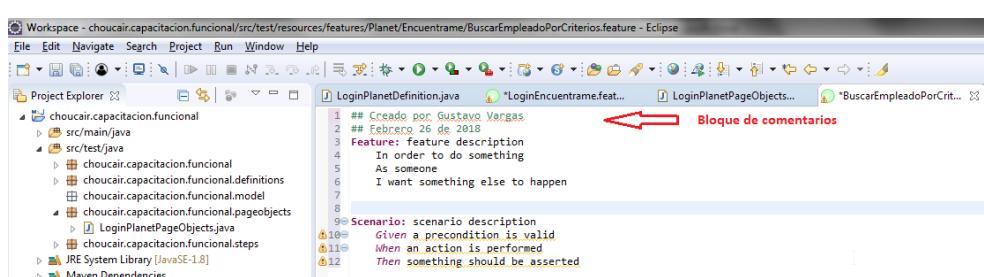
Construyendo el contenido del .feature, seleccionar Feature -



Creando el Scenario Scenario -



El archivo .feature debe visualizarse como indica la siguiente imagen:



Modificando la plantilla generada quedaría como indica la siguiente imagen, donde se puede observar que a los Scenarios se les agregó un tag `@RutaCritica` o `@Smocktest` el cual sirve para identificarlos y poder hacer el llamado desde el RunnerFeature para solo ejecutar los que se requieran:

```

## Creado por Gustavo Vargas
## Febrero 26 de 2018
Feature: Busqueda empleado
Para buscar un empleado
Como usuario registrado
Quiero que sean mostrados los datos en pantalla
@RutaCritica
Scenario: Buscar empleado por criterios
Given Me encuentro logueado con el usuario "gvargas" clave "Choucair.2017"
When entro al directorio de empleados
And Ingreso el nombre "Gustavo" y apellidos "Vargas Avila"
And selecciono pais "<txtPais>" y ciudad "<txtCiudad>"
And Busco empleado
Then Verifico que se visualice el nombre <txtVerificarnombre>
Scenario Outline: Buscar empleado por criterio Nombre
Given Me encuentro logueado con el usuario "gvargas" clave "Choucair.2017"
When entro al directorio de empleados
And Ingreso el nombre <txtNombre> y apellidos <txtApellidos>
And selecciono pais <txtPais> y ciudad <txtCiudad>
And Busco empleado
Then Verifico que se visualice el nombre <txtVerificarnombre>
Examples:
| txtNombre | txtApellidos | txtPais | txtCiudad | txtVerificarnombre |
| "Ericka" | "Caro" | "Colombia" | "Bogota" | "Ericka Lorena" |
| "Flor" | "Toba Vargas" | "Colombia" | "Bogota" | "Flor Angela" |
| "Nidia" | "Corredor Ceballos" | "Colombia" | "Bogota" | "Nidia Estefania" |

```

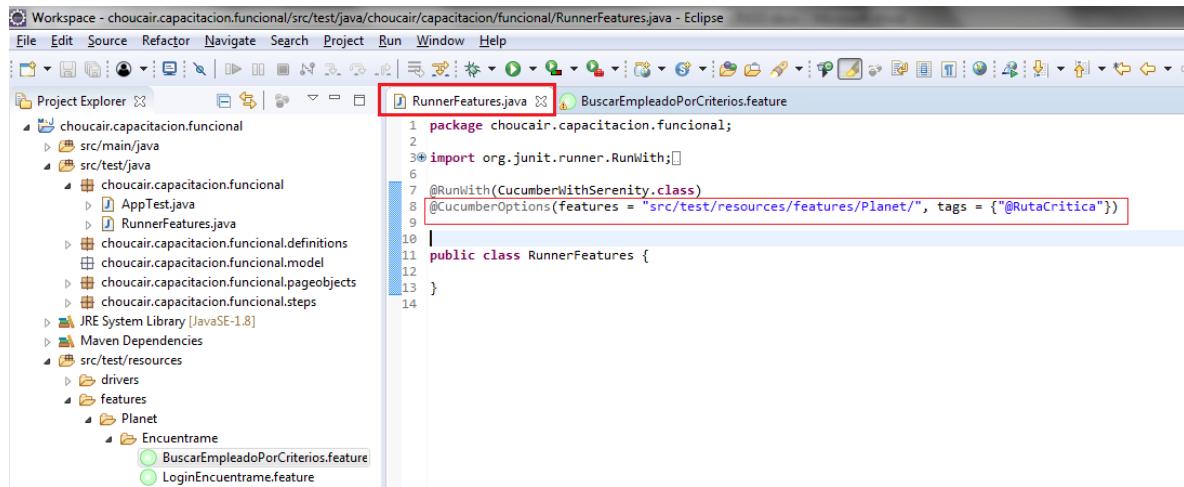
```

## Creado por Gustavo Vargas
## 19-02-2018
Feature: Login Planet
Para ingresar al aplicativo Planet
Como un usuario logueado
Quiero hacer alguna actividad dentro del aplicativo
@Smocktest
Scenario: Logueo en planet con usuario valido
Given Que tengo mis datos de ingreso
When Ingreso el usuario "gvargas"
And Ingreso la contraseña "Choucair.2017"
Then Verifico que ingresé al aplicativo
Scenario Outline: Logueo en planet con usuario valido
Given Que tengo mis datos de ingreso
When Ingreso el usuario <txtNombre>
And Ingreso la contraseña <txtClave>
Then Verifico que ingresé al aplicativo
Examples:
| txtNombre | txtClave |
| "gvargas" | "Choucair.2017" |
| "gvargas" | "Choucair.2017" |

```

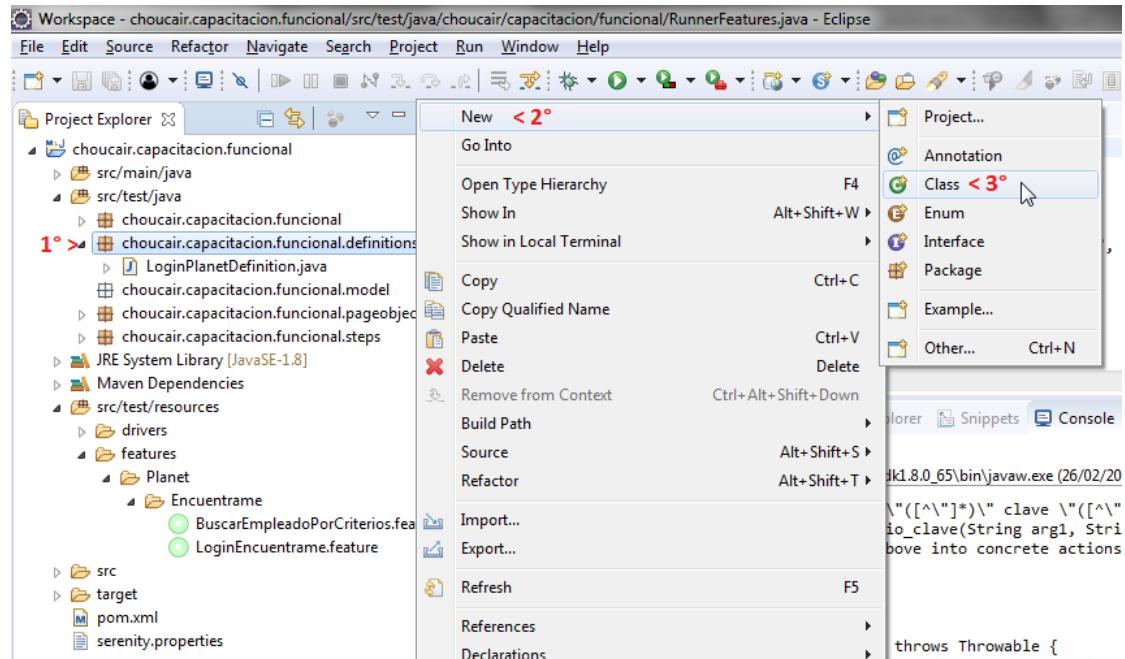
LLAMADO A LOS FEATURE QUE TENGAN TAG ESPECIFICO

Para hacer el llamado al .feature desde la clase RunnerFeature como se indica en la siguiente imagen, con cual se da la instrucción se ejecuten los features que tengan el tag `@RutaCritica`:

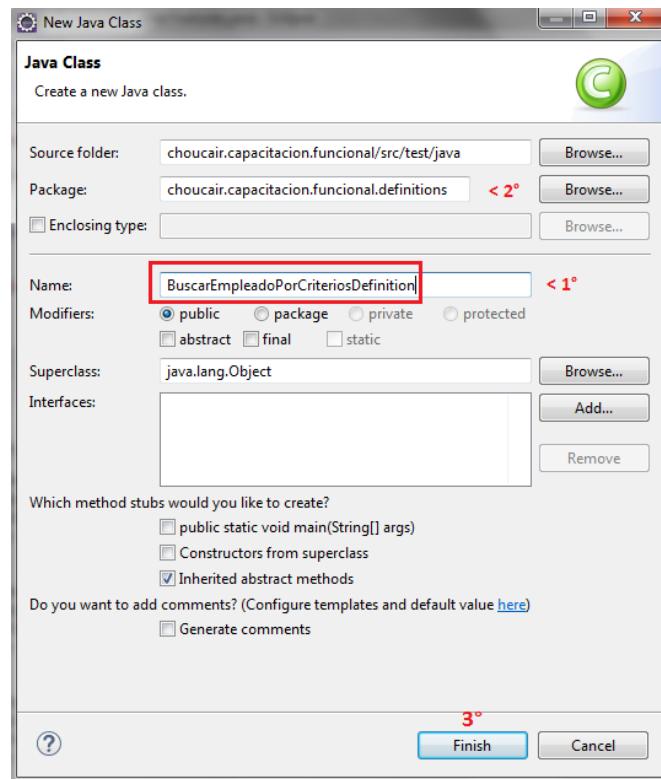


```
1 package choucair.capacitacion.funcional;
2
3 import org.junit.runner.RunWith;
4
5 import io.cucumber.junit.CucumberWithSerenity;
6
7 import io.cucumber.junit.CucumberWithSerenity;
8 import io.cucumber.junit.CucumberWithSerenity;
9
10 public class RunnerFeatures {
11
12 }
13
14
```

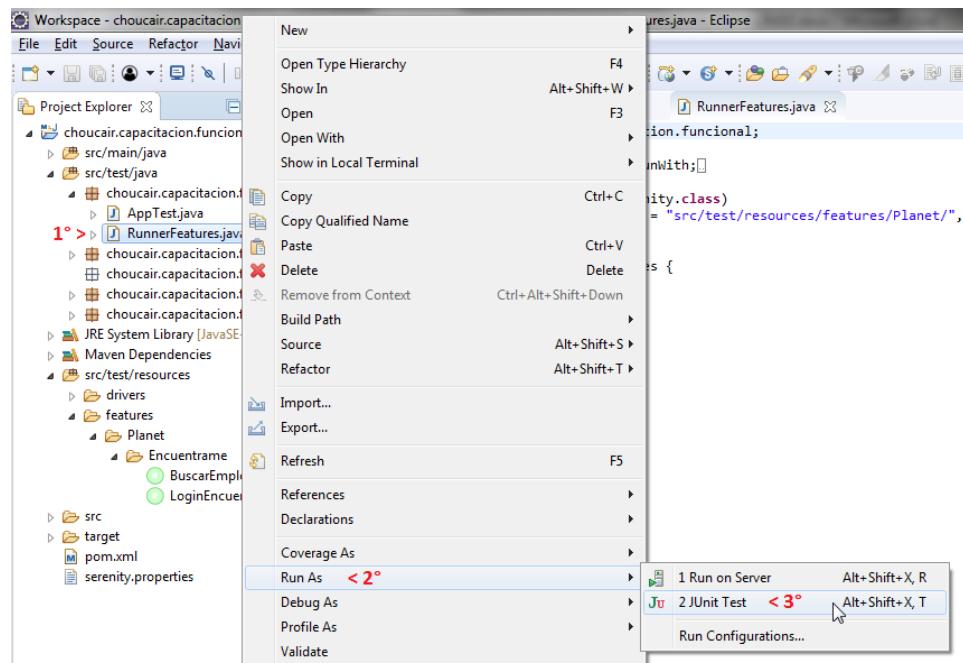
Se crea la clase Definition:



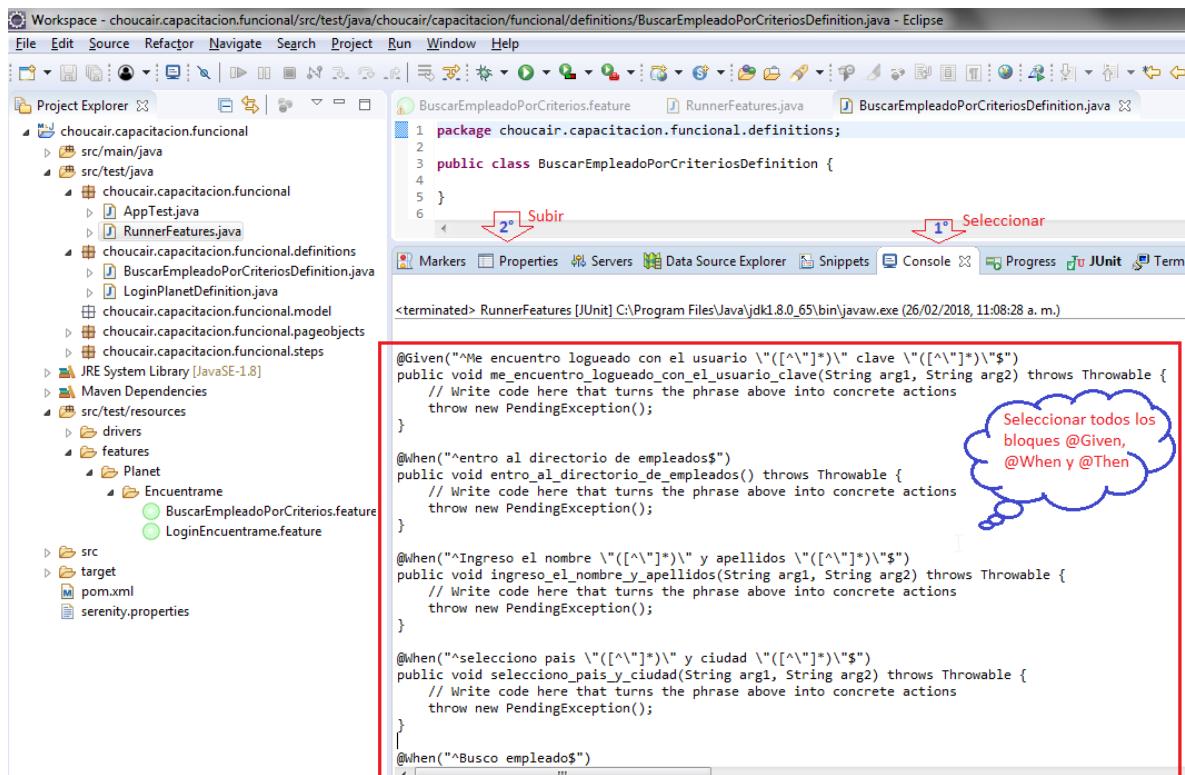
Se le asigna el nombre que debe ser igual al archivo .feature + Definition como indica la siguiente imagen, se debe verificar que se esta creando en el Package correspondiente:



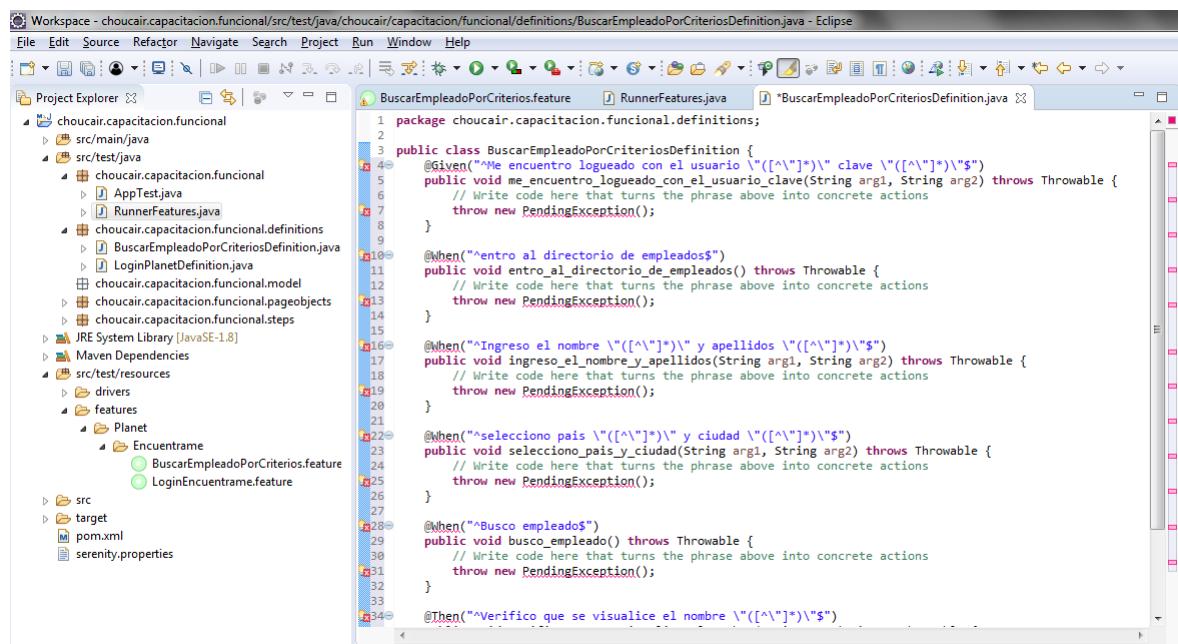
Para ejecutar el RunnerFeature:



Seleccionar todos los @Given, @When y @Then que son generados en la consola, se copian y se pegan en la clase `BuscarEmpleadoPorCriteriosDefinition`



La clase debe verse como indica la siguiente imagen, con los errores que se resuelven como se indica en páginas anteriores:



Se agrega tag Steps y se crea el objeto de BuscarEmpleadoPorCriterios como indica la siguiente imagen, el error se resuelve como se hizo en paginas anteriores:

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/BuscarEmpleadoPorCriteriosDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer BuscarEmpleadoPorCriterios.feature RunnerFeatures.java *BuscarEmpleadoPorCriteriosDefinition.java
1
2 import cucumber.api.java.en.Given;
3 import cucumber.api.java.en.Then;
4 import cucumber.api.java.en.When;
5 import net.thucydides.core.annotations.Steps;
6
7 public class BuscarEmpleadoPorCriteriosDefinition {
8
9     @Steps < 1°
10    BuscarEmpleadoPorCriteriosSteps buscarEmpleadoPorCriteriosSteps; < 2°
11
12
13    @Given("^Me encuentro logueado con el usuario \"([^\"]*)\" clave \"([^\"]*)\"$")
14    public void me_encuentro_logueado_con_el_usuario_clave(String arg1, String arg2) throws Throwable {
15        // Write code here that turns the phrase above into concrete actions
16        //throw new PendingException();
17    }
18
19    @When("^entro al directorio de empleados$")
20    public void entro_al_directorio_de_empleados() throws Throwable {
21        // Write code here that turns the phrase above into concrete actions
22        //throw new PendingException();
23    }
24
25    @When("^Ingreso el nombre \"([^\"]*)\" y apellido \"([^\"]*)\"$")
26    public void ingreso_el_nombre_y_apellido(String arg1, String arg2) throws Throwable {
27        // Write code here that turns the phrase above into concrete actions
28        //throw new PendingException();
29    }
30
31    @When("^selecciono pais \"([^\"]*)\" y ciudad \"([^\"]*)\"$")
32    public void selecciono_pais_y_ciudad(String arg1, String arg2) throws Throwable {
33        // Write code here that turns the phrase above into concrete actions
34        //throw new PendingException();
35    }

```

Se debe crear la clase como indica la siguiente imagen:

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/steps/BuscarEmpleadoPorCriteriosSteps.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer BuscarEmpleadoPorCriterios.feature RunnerFeatures.java *BuscarEmpleadoPorCriteriosDefinition.java BuscarEmpleadoPorCriteriosSteps.java
1 package choucair.capacitacion.funcional.steps;
2
3 public class BuscarEmpleadoPorCriteriosSteps {
4
5 }

```

```

Workspace - choucair.capacitacion.funcional/src/test/java/choucair/capacitacion/funcional/definitions/BuscarEmpleadoPorCriteriosDefinition.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer BuscarEmpleadoPorCriterios.feature *BuscarEmpleadoPorCriteriosDefinition.java BuscarEmpleadoPorCriteriosSteps.java
1
2 import choucair.capacitacion.funcional.steps.BuscadorEmpleadoPorCriteriosSteps;
3 import cucumber.api.java.en.Given;
4 import cucumber.api.java.en.Then;
5 import cucumber.api.java.en.When;
6 import cucumber.api.java.en.When;
7 import net.thucydides.core.annotations.Steps;
8
9 public class BuscarEmpleadoPorCriteriosDefinition {
10
11    @Steps
12    BuscarEmpleadoPorCriteriosSteps buscarEmpleadoPorCriteriosSteps; < Errores solucionados
13
14    @Given("^Me encuentro logueado con el usuario \"([^\"]*)\" clave \"([^\"]*)\"$")
15    public void me_encuentro_logueado_con_el_usuario_clave(String arg1, String arg2) throws Throwable {
16        // Write code here that turns the phrase above into concrete actions
17        //throw new PendingException();
18    }
19

```

Se crea el objeto Steps del LoginPlanetSteps y se soluciona el error como en páginas anteriores:

```

11 import choucair.capacitacion.funcional.steps.BuscarEmpleadoPorCriteriosSteps;
12 @Steps
13 BuscarEmpleadoPorCriteriosSteps buscarEmpleadoPorCriteriosSteps;
14 @Steps
15 LoginPlanetSteps loginPlanetSteps; ← Red arrow here
16
17 @Given("^Me encuentro logueado con el usuario \"([^\"]*)\" clave \"([^\"]*)\"$")
18 public void me_encuentro_logueado_con_el_usuario_clave(String usuario, String clave) throws Throwable {
19     // Write code here that turns the phrase above into concrete actions
20     loginPlanetSteps.LoginPlanet(usuario, clave);
21     //throw new PendingException();
22 }

```

Solucionar este error tiene un tratamiento diferente:

```

11
12 @Steps
13 BuscarEmpleadoPorCriteriosSteps buscarEmpleadoPorCriteriosSteps;
14
15 @Steps
16 LoginPlanetSteps loginPlanetSteps; ← Cursor here
17
18 @Given("^Me encuentro logueado con el usuario \"([^\"]*)\" clave \"([^\"]*)\"$")
19 public void me_encuentro_logueado_con_el_usuario_clave(String usuario, String clave) throws Throwable {
20     // Write code here that turns the phrase above into concrete actions
21     loginPlanetSteps.LoginPlanet(usuario, clave); Posicionar el cursor sobre el error y... ← Red arrow here with '1°'
22     //throw new PendingException();
23 }
24
25 @When("^entro al dire")
26 public void entro_al_dire()
27     // Write code her
28     buscarEmpleadoPor
29     //throw new PendingException();
30 }

```

Se hace el llamado a cada uno de los métodos que deben existir en la clase Definition y el llamado al método de la clase

```

public class BuscarEmpleadoPorCriteriosDefinition {

    @Steps
    BuscarEmpleadoPorCriteriosSteps buscarEmpleadoPorCriteriosSteps;

    @Steps
    LoginPlanetSteps loginPlanetSteps;

    @Given("^Me encuentro logueado con el usuario \"([^\"]*)\" clave \"([^\"]*)\"$")
    public void me_encuentro_logueado_con_el_usuario_clave(String usuario, String clave) throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        loginPlanetSteps.LoginPlanet(usuario, clave);
        //throw new PendingException();
    }

    @When("^entro al directorio de empleados$")
    public void entro_al_directorio_de_empleados() throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        buscarEmpleadoPorCriteriosSteps.IngresoDirectorioEmpleados();
        //throw new PendingException();
    }

    @When("^Ingreso el nombre \"([^\"]*)\" y apellidos \"([^\"]*)\"$")
    public void ingreso_el_nombre_y_apellidos(String nombre, String apellidos) throws Throwable {
        // Write code here that turns the phrase above into concrete actions
        buscarEmpleadoPorCriteriosSteps.IngresarNombresApellidos(nombre, apellidos);
        //throw new PendingException();
    }

    @When("^selecciono pais \"([^\"]*)\" y ciudad \"([^\"]*)\"$")
    public void selecciono_pais_y_ciudad(String pais, String ciudad) throws Throwable {
        // Write code here that turns the phrase above into concrete actions
    }
}

```

```
        buscarEmpleadoPorCriteriosSteps.IngresoPaisCiudad(pais, ciudad);
        //throw new PendingException();
    }

@When("^Busco empleado$")
public void busco_empleado() throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    buscarEmpleadoPorCriteriosSteps.BuscarEmpleado();
    //throw new PendingException();
}

@Then("^Verifico que se visualice el nombre \"([^\"]*)\"$")
public void verifco_que_se_visualice_el_nombre(String nombres) throws Throwable {
    // Write code here that turns the phrase above into concrete actions
    buscarEmpleadoPorCriteriosSteps.VerificarEmpleado(nombres);
    //throw new PendingException();
}

}
```