

EJERCICIO DE AUTOMATIZACIÓN E2E

Autor: Leonardo Páez

URL repositorio: <https://github.com/leopass1997/NTTDataSerenityTest> (el desarrollo se encuentra sobre la rama feature/develop)

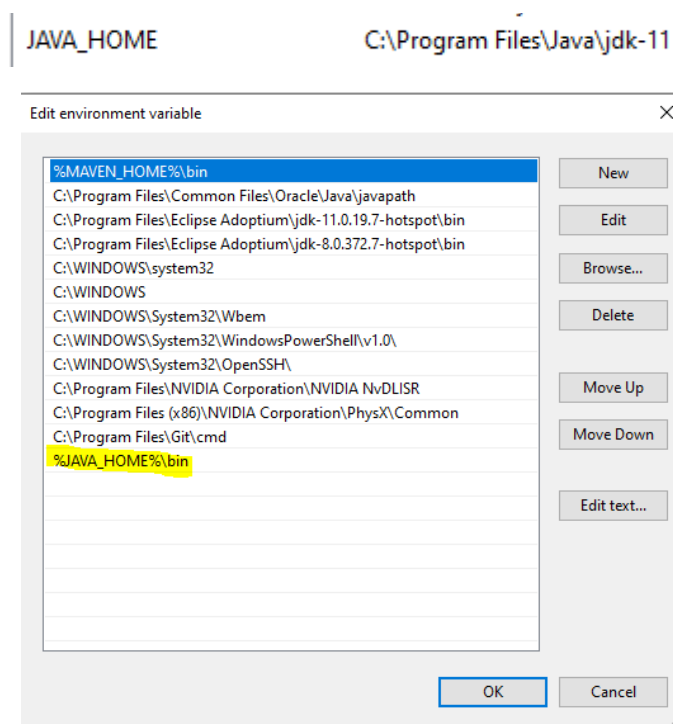
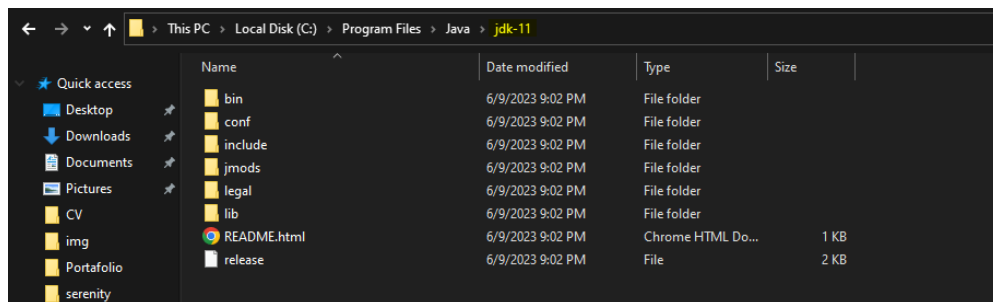
Objetivo

Realizar una prueba funcional automatizada de un flujo de compra en la pagina <https://www.saucedemo.com/> que incluya los siguientes pasos:

- Autenticarse con el **usuario**: standard_user y **password**: secret_sauce
- Agregar dos productos al carrito
- Visualizar el carrito
- Completar el formulario de compra
- Finalizar la compra hasta la confirmación: “THANK YOU FOR YOUR ORDER”

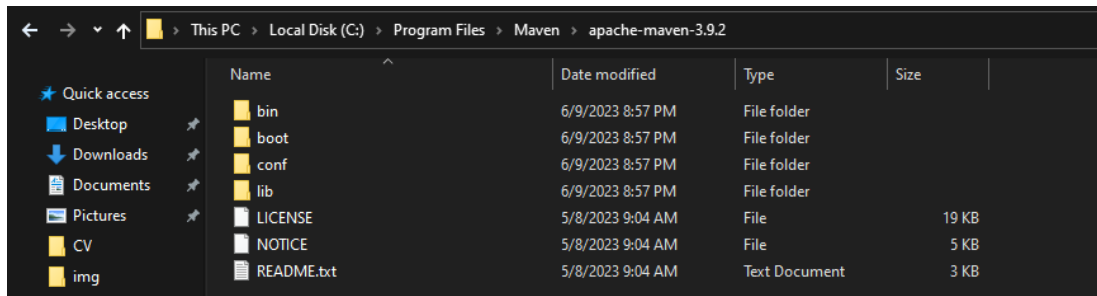
Desarrollo

1. Descargar, instalar y configurar variable de entorno para Java 11



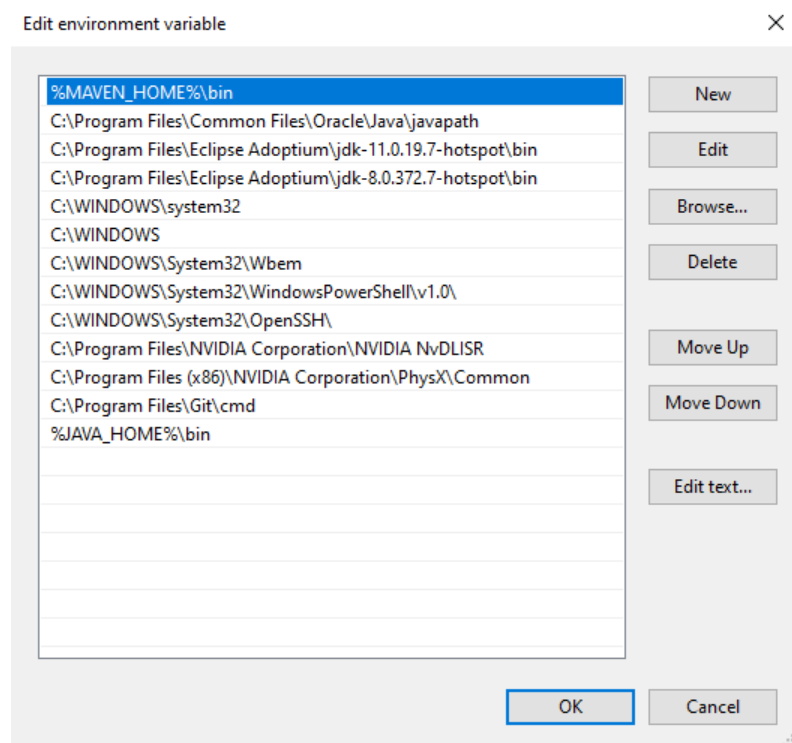
```
C:\Users\LPaez>java -version
java version "11.0.18" 2023-01-17 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.18+9-LTS-195)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.18+9-LTS-195, mixed mode)
```

2. Descargar, instalar y configurar variable de entorno para apache-maven-3.9.2



MAVEN_HOME

C:\Program Files\Maven\apache-maven-3.9.2



```
C:\Users\LPaez>mvn -version
Apache Maven 3.9.2 (c9616018c7a021c1c39be70fb2843d6f5f9b8a1c)
Maven home: C:\Program Files\Maven\apache-maven-3.9.2
Java version: 11.0.19, vendor: Eclipse Adoptium, runtime: C:\Program Files\Eclipse Adoptium\jdk-11.0.19.7-hotspot
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

3. Descargar e Instalar IDE IntelliJ IDEA: <https://www.jetbrains.com/idea/idea/download/#section=windows>
4. Descargar e instalar GIT: <https://git-scm.com/downloads>
5. Revisar documentación Serenity BDD y realizar los pasos sugeridos para generar un proyecto de esqueleto básico de Serenity BDD: https://serenity-bdd.github.io/docs/guide/getting_started

Estaremos utilizando el proyecto SerenityBDD with Cucumber and ScreenPlay.

Getting Started

Creating
Template

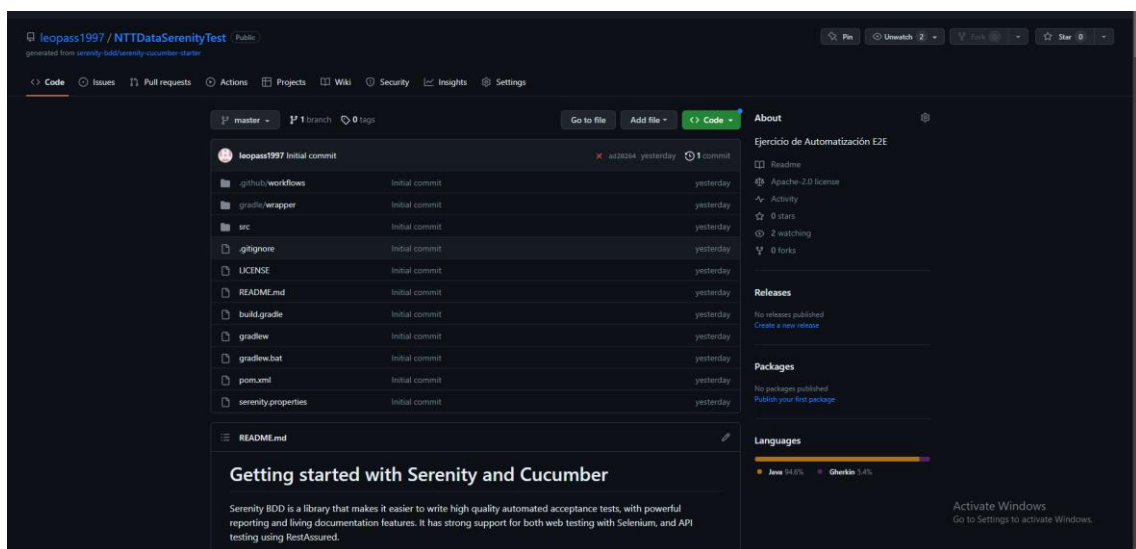
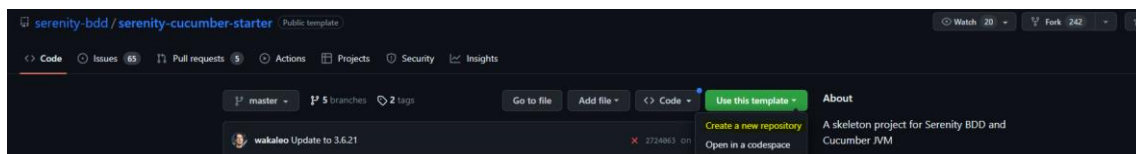
Creating A New Project With The Serenity BDD Templates

The quickest way to get started with a Serenity BDD project is to use one of the starter projects. These are template projects available on Github, that contain a basic Serenity BDD skeleton project and a single test implementation. These template projects are kept up-to-date with the most recent stable version of Serenity BDD.

The main available starter projects are the following:

Template	Description	Get Started
Serenity BDD with JUnit	A simple project structure using Serenity BDD, JUnit 5 and an Action Classes/Page Objects architecture.	Use This Template
Serenity BDD with JUnit and Screenplay	A project built around the Screenplay Pattern using Serenity BDD, JUnit 5 and Serenity Screenplay	Use This Template
Serenity BDD with Cucumber and Screenplay	A BDD-style test suite using Cucumber and Serenity Screenplay	Use This Template

Nos dará la opción para clonarlo en un nuevo repositorio o para abrirlo en un editor de código online. Escogeremos la opción clonarlo en un nuevo repositorio y nos quedara de la siguiente forma. URL repositorio: <https://github.com/leopass1997/NTTDataSerenityTest>



Clonamos el proyecto y generamos una rama **feature/develop** a partir de la rama master. En esta rama es donde se realizará los desarrollos

```
LPaiez@DESKTOP-HCCC7HM MINGW64 ~/Desktop/NTTDataSerenityTest (feature/develop)
$ |
```

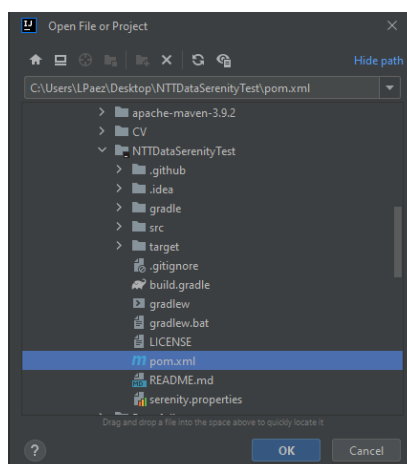
Una vez creada la rama procedemos a ejecutar el siguiente comando `mvn clean package -U`. Este limpiara todo las dependencias y recursos descargados anteriormente por el proyecto.

```
MINGW64/c/Users/LPaez/Desktop/NTTDataSerenityTest
LPaez@DESKTOP-HEC77H MINGW64 ~/Desktop/NTTDataSerenityTest (feature/develop)
$ mvn clean package -u
[INFO] Scanning for projects...
[INFO]
[INFO] ---> net.serenitybdd.starter:cucumber-starter <---
[INFO] Building Sample Serenity BDD project using Cucumber 1.0.0-SNAPSHOT
[INFO] From pom.xml
[INFO] [ jar ]
[INFO] Downloading from central: https://repo.maven.apache.org/maven2/io/cucumber/messages/maven-metadata.xml
Downloaded from central: https://repo.maven.apache.org/maven2/io/cucumber/messages/maven-metadata.xml (2.4 kB at 4.3 kB/s)
[INFO]
[INFO] ---> clean:3.2.0:clean (default-clean) @ cucumber-starter ---
[INFO] Deleting C:\Users\LPaez\Desktop\NTTDataSerenityTest\target
[INFO]
[INFO] ---> resources:3.3.0:resources (default-resources) @ cucumber-starter ---
[INFO] skip non existing resourceDirectory C:\Users\LPaez\Desktop\NTTDataSerenityTest\src\main\resources
[INFO]
[INFO] ---> compiler:3.8.1:compile (default-compile) @ cucumber-starter ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] ---> resources:3.3.0:testResources (default-testResources) @ cucumber-starter ---
[INFO] Copying 3 resources
[INFO]
[INFO] ---> compiler:3.8.1:testCompile (default-testCompile) @ cucumber-starter ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 6 source files to C:\Users\LPaez\Desktop\NTTDataSerenityTest\target\test-classes
[INFO]
[INFO] ---> surefire:3.0.0-M5:test (default-test) @ cucumber-starter ---
[INFO] Tests are skipped.
[INFO]
[INFO] ---> jar:3.3.0:jar (default-jar) @ cucumber-starter ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\Users\LPaez\Desktop\NTTDataSerenityTest\target\cucumber-starter-1.0.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.538 s
[INFO] Finished at: 2023-06-11T01:15:07-05:00
[INFO]
[WARNING] Plugin validation issues were detected in 3 plugin(s)
[WARNING]
[WARNING] * org.apache.maven.plugins:maven-compiler-plugin:3.8.1
[WARNING] * org.apache.maven.plugins:maven-surefire-plugin:3.0.0-M5
[WARNING] * org.apache.maven.plugins:maven-resources-plugin:3.3.0
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case insensitive): [BRIEF, DEFAULT, VERBOSE]
```

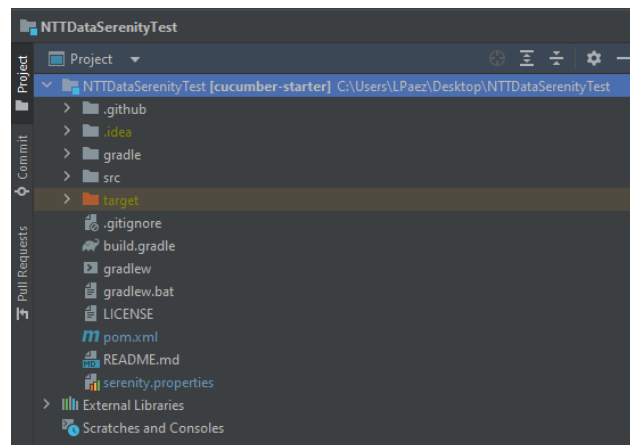
Procedemos a ejecutar el siguiente comando `mvn clean install -DskipTests`. Este hará un install sobre el proyecto sin ejecutar los tests y verificará que no exista errores y comenzará a descargar las dependencias y recursos que el proyecto utilizará a partir de su `pom.xml`.

```
MINGW64/c/Users/LPaez/Desktop/NTTDataSerenityTest (feature/develop)
$ mvn clean install -DskipTests
[INFO] Scanning for projects...
[INFO]
[INFO] ---> net.serenitybdd.starter:cucumber-starter <---
[INFO] Building Sample Serenity BDD project using Cucumber 1.0.0-SNAPSHOT
[INFO] From pom.xml
[INFO] [ jar ]
[INFO] ---> clean:3.2.0:clean (default-clean) @ cucumber-starter ---
[INFO] Deleting C:\Users\LPaez\Desktop\NTTDataSerenityTest\target
[INFO]
[INFO] ---> resources:3.3.0:resources (default-resources) @ cucumber-starter ---
[INFO] skip non existing resourceDirectory C:\Users\LPaez\Desktop\NTTDataSerenityTest\src\main\resources
[INFO]
[INFO] ---> compiler:3.8.1:compile (default-compile) @ cucumber-starter ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] ---> resources:3.3.0:testResources (default-testResources) @ cucumber-starter ---
[INFO] Copying 3 resources
[INFO]
[INFO] ---> compiler:3.8.1:testCompile (default-testCompile) @ cucumber-starter ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 6 source files to C:\Users\LPaez\Desktop\NTTDataSerenityTest\target\test-classes
[INFO]
[INFO] ---> surefire:3.0.0-M5:test (default-test) @ cucumber-starter ---
[INFO] Tests are skipped.
[INFO]
[INFO] ---> jar:3.3.0:jar (default-jar) @ cucumber-starter ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\Users\LPaez\Desktop\NTTDataSerenityTest\target\cucumber-starter-1.0.0-SNAPSHOT.jar
[INFO]
[INFO] ---> failsafe:3.0.0-M5:integration-test (default) @ cucumber-starter ---
[INFO] Tests are skipped.
[INFO]
[INFO] ---> surefire:3.0.0-M5:test (default-test) @ cucumber-starter ---
[INFO] Tests are skipped.
[INFO]
[INFO] ---> failsafe:3.0.0-M5:verify (default) @ cucumber-starter ---
[INFO] Tests are skipped.
[INFO]
[INFO] ---> install:3.0.0:install (default-install) @ cucumber-starter ---
[INFO] Installing C:\Users\LPaez\Desktop\NTTDataSerenityTest\pom.xml to C:\Users\LPaez\AppData\Local\repository\net\serenitybdd\starter\cucumber-starter\1.0.0-SNAPSHOT\pom
[INFO] Installing C:\Users\LPaez\Desktop\NTTDataSerenityTest\target\cucumber-starter-1.0.0-SNAPSHOT.jar to C:\Users\LPaez\AppData\Local\repository\net\serenitybdd\starter\cucumber-starter\1.0.0-SNAPSHOT\cucumber-starter-1.0.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 5.578 s
[INFO] Finished at: 2023-06-11T01:17:09-05:00
[INFO]
[WARNING] Plugin validation issues were detected in 4 plugin(s)
[WARNING]
[WARNING] * org.apache.maven.plugins:maven-failsafe-plugin:3.0.0-M5
[WARNING] * org.apache.maven.plugins:maven-compiler-plugin:3.8.1
[WARNING] * org.apache.maven.plugins:maven-surefire-plugin:3.0.0-M5
[WARNING] * org.apache.maven.plugins:maven-resources-plugin:3.3.0
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case insensitive): [BRIEF, DEFAULT, VERBOSE]
```

Abriremos el proyecto en IntelliJ IDEA a partir desde su `pom.xml`. Esto permitirá que el proyecto se compile dentro del IDE y también que descargue dependencias.



Una vez adentro nos generara la siguiente estructura.



Abriremos el archivo pom.xml y observaremos que para nuestro proyecto estaremos utilizando la versión 3.6.21 de serenity.

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <serenity.version>3.6.21</serenity.version>
  <encoding>UTF-8</encoding>
  <tags></tags>
  <webdriver.base.url></webdriver.base.url>
</properties>
```

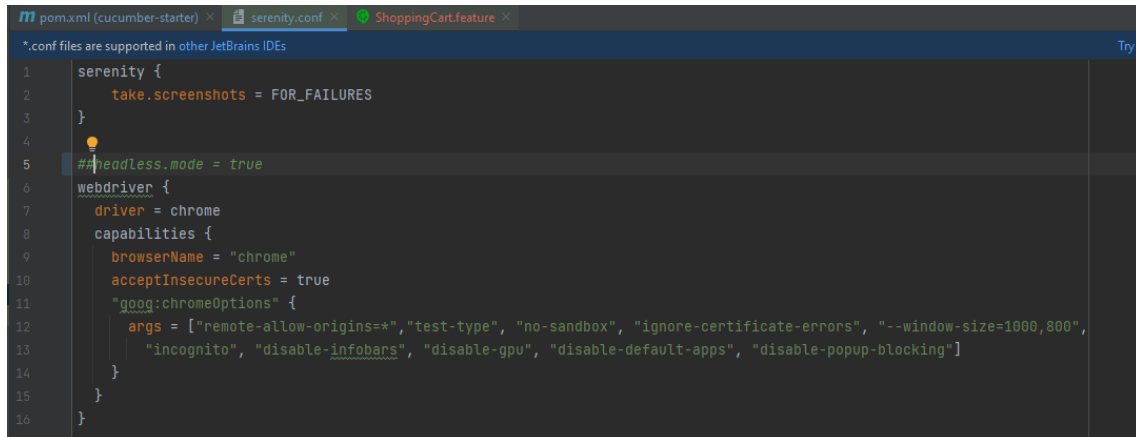
Adicionalmente cuando ejecutemos el comando `mvn clean verify -Dcucumber.filter.tags "` esto nos permitirá generar el reporte automatico de serenity reports.

```
<plugin>
  <groupId>net.serenity-bdd.maven.plugins</groupId>
  <artifactId>serenity-maven-plugin</artifactId>
  <version>${serenity.version}</version>
  <configuration>
    <tags>${tags}</tags>
  </configuration>
  <executions>
    <execution>
      <id>serenity-reports</id>
      <phase>post-integration-test</phase>
      <goals>
        <goal>aggregate</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Finalmente se realiza la implementación del webdriver para no depender de actualizaciones manuales dentro del proyecto y que todas las actualizaciones se hagan automáticas a partir del serenity.conf.

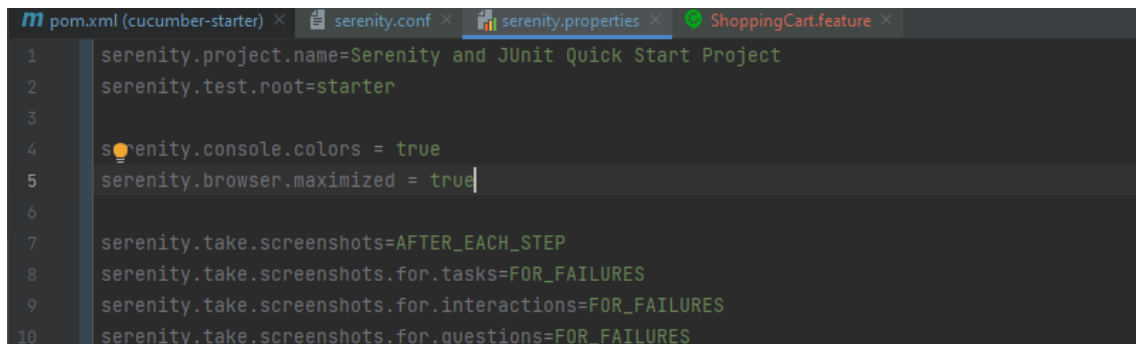
```
<dependency>
  <groupId>net.serenity-bdd</groupId>
  <artifactId>serenity-screenplay-webdriver</artifactId>
  <version>${serenity.version}</version>
  <scope>test</scope>
</dependency>
```

En nuestro serenity.conf tenemos la siguiente configuración. Tenemos el webdriver apuntando al driver de Chrome y una configuración inicial para cuando se vaya a ejecutar nuestras pruebas en este navegador.

A screenshot of an IDE showing the configuration file serenity.conf. The file is open in a tab alongside pom.xml (cucumber-starter) and ShoppingCart.feature. The code defines a serenity block with take.screenshots = FOR_FAILURES, and a webdriver block with driver = chrome and a list of capabilities for Chrome, including browserName, acceptInsecureCerts, and various chromeOptions like remote-allow-origins, no-sandbox, and window-size.

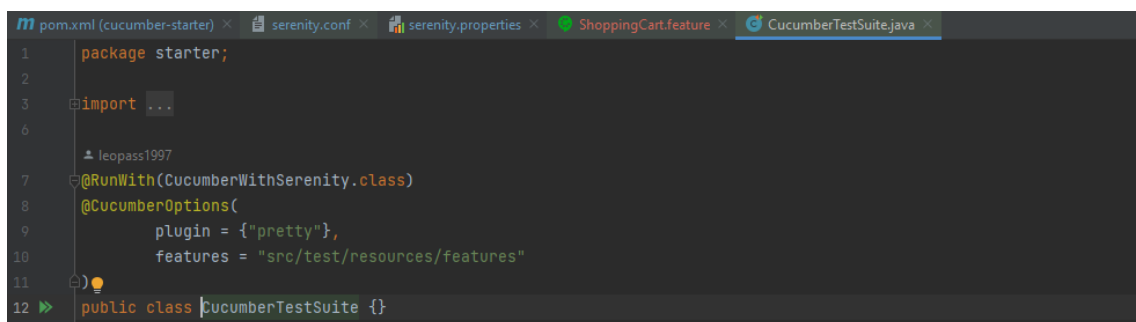
```
1 serenity {
2   take.screenshots = FOR_FAILURES
3 }
4
5 ##headless.mode = true
6 webdriver {
7   driver = chrome
8   capabilities {
9     browserName = "chrome"
10    acceptInsecureCerts = true
11    "goog:chromeOptions" {
12      args = ["remote-allow-origins*", "test-type", "no-sandbox", "ignore-certificate-errors", "--window-size=1000,800",
13             "incognito", "disable-infobars", "disable-gpu", "disable-default-apps", "disable-popup-blocking"]
14    }
15  }
16 }
```

En nuestro serenity.properties tenemos la configuración inicial para maximizar la ventana del browser cuando se este ejecutando y las condiciones para la toma de capturas de nuestro reporte de serenity.

A screenshot of an IDE showing the configuration file serenity.properties. The file is open in a tab alongside pom.xml (cucumber-starter), serenity.conf, and ShoppingCart.feature. The code sets project and test root paths, console colors, browser maximization, and screenshot-taking rules for tasks, interactions, and questions.

```
1 serenity.project.name=Serenity and JUnit Quick Start Project
2 serenity.test.root=starter
3
4 serenity.console.colors = true
5 serenity.browser.maximized = true
6
7 serenity.take.screenshots=AFTER_EACH_STEP
8 serenity.take.screenshots.for.tasks=FOR_FAILURES
9 serenity.take.screenshots.for.interactions=FOR_FAILURES
10 serenity.take.screenshots.for.questions=FOR_FAILURES
```

En nuestro cucumbertestsuit.java tenemos la referencia apuntando a la siguiente dirección para que tome de ahí los features cuando vaya a buscarlos.

A screenshot of an IDE showing the CucumberTestSuite.java file. The file is open in a tab alongside pom.xml (cucumber-starter), serenity.conf, serenity.properties, and ShoppingCart.feature. The code defines a package, imports, and a class with annotations for running tests with Serenity and Cucumber, specifying the features directory.

```
1 package starter;
2
3 import ...
4
5
6
7 @RunWith(CucumberWithSerenity.class)
8 @CucumberOptions(
9   plugin = {"pretty"},
10  features = "src/test/resources/features"
11 )
12 public class CucumberTestSuite {}
```

A continuación, armamos nuestro feature llamado ShoppingCart el cual hace referencia al flujo de compras que vamos a realizar dentro de la web mencionada. Este escenario de prueba es armado con Serenity BDD y Cucumber. Cucumber prácticamente es un analizador de lenguaje gherkin, el lenguaje de nuestro caso de prueba.

Observaciones feature:

- De la línea 1 a la 4 se están colocando información general sobre el lenguaje del feature, desarrollador, aprobador funcional y el comando para ejecutar el caso de prueba desde el git bash.
- El tag @FeatureCompletoShoppingCart puede ser utilizado para correr todos los escenarios dentro del feature en caso de tener más de un escenario.
- Se brinda una característica y descripción breve de lo que va a tratar el feature y sus escenarios de prueba.
- El tag @FlujoComprasProductos es el tag del escenario. Al tener un único caso este será el que estaremos utilizando para correr el escenario y generar el reporte de serenity.
- En este caso el escenario se trata de un escenario outline o un esquema del escenario. Esto quiere decir que este escenario en específico recibirá data desde la sección ejemplos que observamos al final.

```
1 #language: es
2 #Automatizador: Leonardo
3 #Funcional: NTTData
4 #Comando Ejecutor: mvn clean verify -Dcucumber.filter.tags="@FlujoComprasProductos:"
5
6 @FeatureCompletoShoppingCart
7 Característica: Compra de Productos en Swag Labs
8
9 COMO usuario de la plataforma Swag Labs
10 QUIERO realizar la compra de productos
11 PARA evidenciar la compra correcta de productos online
12
13 @FlujoComprasProductos
14 Esquema del escenario: Flujo de compras dentro de Swag Labs
15 Dado Leonardo ingresa a la aplicación Swag Labs
16 Cuando ingresa <usuario> y <password> en formulario de login
17 Y hace clic en boton Login
18 Y hace clic en boton agregar a carrito de <productoUno> y <productoDos>
19 Y se hace clic en boton carrito de compras para visualizar los productos agregados
20 Y se hace clic en boton verificar
21 Y se ingresa <nombre>, <apellido> y <codigoPostal> del comprador, se hace clic en boton continuar
22 Entonces se hace clic en boton finalizar, se valida compra exitosa, se regresa a pagina principal
23
24 Ejemplos:
25 | usuario | password | productoUno | productoDos | nombre | apellido | codigoPostal |
26 | "standard_user" | "secret_sauce" | "Sauce Labs Backpack" | "Test.allTheThings() T-Shirt (Red)" | "Leonardo" | "Paez" | "171168" |
```

Con nuestro feature ya finalizado únicamente nos queda generar el stepdefinition para desarrollar la funcionalidad solicitada.

Observaciones stepdefinition:

- El nombre del StepDefinition es en relación al nombre del feature que vamos a desarrollar la funcionalidad.
- El stepdefinition prácticamente viene a ser una traducción de su respectivo feature. Donde cada método establecido en código dentro de la clase java, sería una línea correspondiente al caso de prueba.
- Dentro del stepdefinition se han realizado desarrollos con las siguientes características: mapeos manuales(xpath), id de elementos, acciones de serenity bdd, aserciones, targets, webelements.
- Para garantizar el funcionamiento correcto después de realizar una acción como dar clic, llenar un valor o ver un mensaje, se han realizado varias validaciones a nivel de código con acciones propias de serenity bdd.

```

public class ShoppingCartStepDefinition {

    @Cuando("ingresa {string} y {string} en formulario de login")
    public void ingresa_y_en_formulario_de_login(String usuario, String contrasena) {
        theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Displayed.of
            (Target.the( targetElementName: "Username").locatedBy( cssOrXPathSelector: "//input[@id='user-name']")))).isTrue());
        theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Displayed.of
            (Target.the( targetElementName: "Password").locatedBy( cssOrXPathSelector: "//input[@id='password']")))).isTrue());
        theActorInTheSpotlight().attemptsTo(Enter.theValue(usuario).into
            (Target.the( targetElementName: "Username").locatedBy( cssOrXPathSelector: "//input[@id='user-name']")));
        theActorInTheSpotlight().attemptsTo(Enter.theValue(contrasena).into
            (Target.the( targetElementName: "Password").locatedBy( cssOrXPathSelector: "//input[@id='password']")));
    }

    @Cuando("hace clic en boton Login")
    public void hace_clic_en_boton_login() {
        theActorInTheSpotlight().attemptsTo(Click.on
            (Target.the( targetElementName: "Login").located(By.id("login-button"))));
        Target inventory = Target.the( targetElementName: "Inventario").located(By.xpath
            ( xpathExpression: "//div[@id='inventory_container' and @class='inventory_container']"));
        theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Enabled.of(inventory))).
            isTrue());
    }
}

```

```

@Cuando("hace clic en boton agregar a carrito de {string} y {string}")
public void hace_clic_en_boton_agregar_a_carrito_de_y(String productoUno, String productoDos) {
    Target producto1 = Target.the( targetElementName: "Producto Uno Add").located(By.xpath
        ( xpathExpressions: "//div[contains(text(),' + productoUno + ')]" +
            "//ancestor::div[@class='inventory_item_description']/button[contains(text(),'Add to cart']"));
    Target producto2 = Target.the( targetElementName: "Producto Dos Add").located(By.xpath
        ( xpathExpressions: "//div[contains(text(),' + productoDos + ')]" +
            "//ancestor::div[@class='inventory_item_description']/button[contains(text(),'Add to cart']"));
    theActorInTheSpotlight().attemptsTo(Click.on(producto1));
    theActorInTheSpotlight().attemptsTo(Click.on(producto2));
    producto1 = Target.the( targetElementName: "Producto Uno Remove").located(By.xpath
        ( xpathExpressions: "//div[contains(text(),' + productoUno + ')]" +
            "//ancestor::div[@class='inventory_item_description']/button[contains(text(),'Remove']"));
    producto2 = Target.the( targetElementName: "Producto Dos Remove").located(By.xpath
        ( xpathExpressions: "//div[contains(text(),' + productoDos + ')]" +
            "//ancestor::div[@class='inventory_item_description']/button[contains(text(),'Remove']"));
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Displayed.of(producto1))).isTrue());
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Displayed.of(producto2))).isTrue());
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Text.of(By.
        xpath( xpathExpression: "//div[@id='shopping_cart_container']/span[@class='shopping_cart_badge']")))).
        isNotEqualTo( expected: "0")));
}

@Cuando("se hace clic en boton carrito de compras para visualizar los productos agregados")
public void se_hace_clic_en_boton_carrito_de_compras_para_visualizar_los_productos_agregados() {
    Target carritoCompras = Target.the( targetElementName: "Carrito Compras").located(By.xpath
        ( xpathExpressions: "//div[@id='shopping_cart_container']/a[@class='shopping_cart_link']"));
    theActorInTheSpotlight().attemptsTo(Click.on(carritoCompras));
    WebElement gridProductosCarro = BrowseTheWeb.as(theActorInTheSpotlight()).$(By.
        xpath( xpathExpression: "//div[@class='cart_list']"));
    List<WebElement> cantidadProducto = gridProductosCarro.findElements(By.
        xpath( xpathExpression: "//div[@class='cart_item']"));
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Text.of(By.
        xpath( xpathExpression: "//div[@id='shopping_cart_container']/span[@class='shopping_cart_badge']")))).
        isEqualTo(String.valueOf(cantidadProducto.size())));
}

```



```

@Cuando("se hace clic en boton verificar")
public void se_hace_clic_en_boton_verificar() {
    theActorInTheSpotlight().attemptsTo(Click.on
        (Target.the(targetElementName: "Verificar").located(By.id("checkout"))));
}

@Cuando("se ingresa {string}, {string} y {string} del comprador, se hace clic en boton continuar")
public void se_ingresa_y_del_comprador_se_hace_clic_en_boton_continuar(String nombre, String apellido, String codigoPostal) {
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Displayed.of
        (Target.the(targetElementName: "Primer Nombre").locatedBy(cssOrXPathSelector: "//input[@id='first-name']")))).isTrue());
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Displayed.of
        (Target.the(targetElementName: "Apellido").locatedBy(cssOrXPathSelector: "//input[@id='last-name']")))).isTrue());
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Displayed.of
        (Target.the(targetElementName: "Codigo Postal").locatedBy(cssOrXPathSelector: "//input[@id='postal-code']")))).isTrue());
    theActorInTheSpotlight().attemptsTo(Enter.theValue(nombre).into
        (Target.the(targetElementName: "Primer Nombre").locatedBy(cssOrXPathSelector: "//input[@id='first-name']")));
    theActorInTheSpotlight().attemptsTo(Enter.theValue(apellido).into
        (Target.the(targetElementName: "Apellido").locatedBy(cssOrXPathSelector: "//input[@id='last-name']")));
    theActorInTheSpotlight().attemptsTo(Enter.theValue(codigoPostal).into
        (Target.the(targetElementName: "Codigo Postal").locatedBy(cssOrXPathSelector: "//input[@id='postal-code']")));
    theActorInTheSpotlight().attemptsTo(Click.on
        (Target.the(targetElementName: "Continuar").located(By.id("continue"))));
}

@Entonces("se hace clic en boton finalizar, se valida compra exitosa, se regresa a pagina principal")
public void se_hace_clic_en_boton_finalizar_se_valida_compra_exitosa_se_regresa_a_pagina_principal() {
    theActorInTheSpotlight().attemptsTo(Click.on
        (Target.the(targetElementName: "Finalizar Compra").located(By.id("finish"))));
    theActorInTheSpotlight().attemptsTo(Ensure.that(theActorInTheSpotlight().asksFor(Text.of(By.
        xpath(xpathExpression: "//h2[@class='complete-header']")))).
        isEqualTo(expected: "Thank you for your order!"));
    theActorInTheSpotlight().attemptsTo(Click.on
        (Target.the(targetElementName: "Back Home").located(By.id("back-to-product"))));
}

```

Finalmente ya implementado el feature y el stepdefinition para dar vida a nuestro escenario de prueba, se abre una terminal ya pueda ser dentro el IDE o desde el git y ejecutaremos el siguiente comando `mvn clean verify -Dcucumber.filter.tags="@FlujoComprasProductos"`. Este comando es el mismo que se coloco al inicio en los comentarios del feature. Cuando se ejecute se abrirá el navegador establecido en el `serenity.conf` y empezara la ejecución automática, dejando como resultado final el serenity report para validar como fue la ejecución.

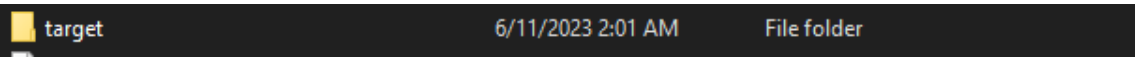
The screenshot displays an IDE environment where a Cucumber test is being executed. The top panel shows the project structure with 'ShoppingCartDefinition' and 'ShoppingCartFeature'. The middle panel shows the test code for 'FlujoComprasProductos'. The bottom panel shows the terminal output of the command `mvn clean verify -Dcucumber.filter.tags="@FlujoComprasProductos"`, which includes a Cucumber report and a link to the Serenity report.

Terminal Output:

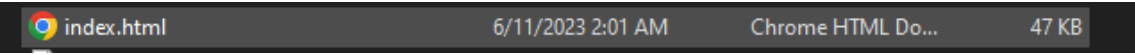
```

02:01:29.183 [main] INFO -
[+] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109] [110] [111] [112] [113] [114] [115] [116] [117] [118] [119] [120] [121] [122] [123] [124] [125] [126] [127] [128] [129] [130] [131] [132] [133] [134] [135] [136] [137] [138] [139] [140] [141] [142] [143] [144] [145] [146] [147] [148] [149] [150] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160] [161] [162] [163] [164] [165] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183] [184] [185] [186] [187] [188] [189] [190] [191] [192] [193] [194] [195] [196] [197] [198] [199] [200] [201] [202] [203] [204] [205] [206] [207] [208] [209] [210] [211] [212] [213] [214] [215] [216] [217] [218] [219] [220] [221] [222] [223] [224] [225] [226] [227] [228] [229] [230] [231] [232] [233] [234] [235] [236] [237] [238] [239] [240] [241] [242] [243] [244] [245] [246] [247] [248] [249] [250] [251] [252] [253] [254] [255] [256] [257] [258] [259] [260] [261] [262] [263] [264] [265] [266] [267] [268] [269] [270] [271] [272] [273] [274] [275] [276] [277] [278] [279] [280] [281] [282] [283] [284] [285] [286] [287] [288] [289] [290] [291] [292] [293] [294] [295] [296] [297] [298] [299] [300] [301] [302] [303] [304] [305] [306] [307] [308] [309] [310] [311] [312] [313] [314] [315] [316] [317] [318] [319] [320] [321] [322] [323] [324] [325] [326] [327] [328] [329] [330] [331] [332] [333] [334] [335] [336] [337] [338] [339] [340] [341] [342] [343] [344] [345] [346] [347] [348] [349] [350] [351] [352] [353] [354] [355] [356] [357] [358] [359] [360] [361] [362] [363] [364] [365] [366] [367] [368] [369] [370] [371] [372] [373] [374] [375] [376] [377] [378] [379] [380] [381] [382] [383] [384] [385] [386] [387] [388] [389] [390] [391] [392] [393] [394] [395] [396] [397] [398] [399] [400] [401] [402] [403] [404] [405] [406] [407] [408] [409] [410] [411] [412] [413] [414] [415] [416] [417] [418] [419] [420] [421] [422] [423] [424] [425] [426] [427] [428] [429] [430] [431] [432] [433] [434] [435] [436] [437] [438] [439] [440] [441] [442] [443] [444] [445] [446] [447] [448] [449] [450] [451] [452] [453] [454] [455] [456] [457] [458] [459] [460] [461] [462] [463] [464] [465] [466] [467] [468] [469] [470] [471] [472] [473] [474] [475] [476] [477] [478] [479] [480] [481] [482] [483] [484] [485] [486] [487] [488] [489] [490] [491] [492] [493] [494] [495] [496] [497] [498] [499] [500] [501] [502] [503] [504] [505] [506] [507] [508] [509] [510] [511] [512] [513] [514] [515] [516] [517] [518] [519] [520] [521] [522] [523] [524] [525] [526] [527] [528] [529] [530] [531] [532] [533] [534] [535] [536] [537] [538] [539] [540] [541] [542] [543] [544] [545] [546] [547] [548] [549] [550] [551] [552] [553] [554] [555] [556] [557] [558] [559] [560] [561] [562] [563] [564] [565] [566] [567] [568] [569] [570] [571] [572] [573] [574] [575] [576] [577] [578] [579] [580] [581] [582] [583] [584] [585] [586] [587] [588] [589] [590] [591] [592] [593] [594] [595] [596] [597] [598] [599] [600] [601] [602] [603] [604] [605] [606] [607] [608] [609] [610] [611] [612] [613] [614] [615] [616] [617] [618] [619] [620] [621] [622] [623] [624] [625] [626] [627] [628] [629] [630] [631] [632] [633] [634] [635] [636] [637] [638] [639] [640] [641] [642] [643] [644] [645] [646] [647] [648] [649] [650] [651] [652] [653] [654] [655] [656] [657] [658] [659] [660] [661] [662] [663] [664] [665] [666] [667] [668] [669] [670] [671] [672] [673] [674] [675] [676] [677] [678] [679] [680] [681] [682] [683] [684] [685] [686] [687] [688] [689] [690] [691] [692] [693] [694] [695] [696] [697] [698] [699] [700] [701] [702] [703] [704] [705] [706] [707] [708] [709] [710] [711] [712] [713] [714] [715] [716] [717] [718] [719] [720] [721] [722] [723] [724] [725] [726] [727] [728] [729] [730] [731] [732] [733] [734] [735] [736] [737] [738] [739] [740] [741] [742] [743] [744] [745] [746] [747] [748] [749] [750] [751] [752] [753] [754] [755] [756] [757] [758] [759] [760] [761] [762] [763] [764] [765] [766] [767] [768] [769] [770] [771] [772] [773] [774] [775] [776] [777] [778] [779] [780] [781] [782] [783] [784] [785] [786] [787] [788] [789] [790] [791] [792] [793] [794] [795] [796] [797] [798] [799] [800] [801] [802] [803] [804] [805] [806] [807] [808] [809] [810] [811] [812] [813] [814] [815] [816] [817] [818] [819] [820] [821] [822] [823] [824] [825] [826] [827] [828] [829] [830] [831] [832] [833] [834] [835] [836] [837] [838] [839] [840] [841] [842] [843] [844] [845] [846] [847] [848] [849] [850] [851] [852] [853] [854] [855] [856] [857] [858] [859] [860] [861] [862] [863] [864] [865] [866] [867] [868] [869] [870] [871] [872] [873] [874] [875] [876] [877] [878] [879] [880] [881] [882] [883] [884] [885] [886] [887] [888] [889] [890] [891] [892] [893] [894] [895] [896] [897] [898] [899] [900] [901] [902] [903] [904] [905] [906] [907] [908] [909] [910] [911] [912] [913] [914] [915] [916] [917] [918] [919] [920] [921] [922] [923] [924] [925] [926] [927] [928] [929] [930] [931] [932] [933] [934] [935] [936] [937] [938] [939] [940] [941] [942] [943] [944] [945] [946] [947] [948] [949] [950] [951] [952] [953] [954] [955] [956] [957] [958] [959] [960] [961] [962] [963] [964] [965] [966] [967] [968] [969] [970] [971] [972] [973] [974] [975] [976] [977] [978] [979] [980] [981] [982] [983] [984] [985] [986] [987] [988] [989] [990] [991] [992] [993] [994] [995] [996] [997] [998] [999] [1000] [1001] [1002] [1003] [1004] [1005] [1006] [1007] [1008] [1009] [1010] [1011] [1012] [1013] [1014] [1015] [1016] [1017] [1018] [1019] [1020] [1021] [1022] [1023] [1024] [1025] [1026] [1027] [1028] [1029] [1030] [1031] [1032] [1033] [1034] [1035] [1036] [1037] [1038] [1039] [1040] [1041] [1042] [1043] [1044] [1045] [1046] [1047] [1048] [1049] [1050] [1051] [1052] [1053] [1054] [1055] [1056] [1057] [1058] [1059] [1060] [1061] [1062] [1063] [1064] [1065] [1066] [1067] [1068] [1069] [1070] [1071] [1072] [1073] [1074] [1075] [1076] [1077] [1078] [1079] [1080] [1081] [1082] [1083] [1084] [1085] [1086] [1087] [1088] [1089] [1090] [1091] [1092] [1093] [1094] [1095] [1096] [1097] [1098] [1099] [1100] [1101] [1102] [1103] [1104] [1105] [1106] [1107] [1108] [1109] [1110] [1111] [1112] [1113] [1114] [1115] [1116] [1117] [1118] [1119] [1120] [1121] [1122] [1123] [1124] [1125] [1126] [1127] [1128] [1129] [1130] [1131] [1132] [1133] [1134] [1135] [1136] [1137] [1138] [1139] [1140] [1141] [1142] [1143] [1144] [1145] [1146] [1147] [1148] [1149] [1150] [1151] [1152] [1153] [1154] [1155] [1156] [1157] [1158] [1159] [1160] [1161] [1162] [1163] [1164] [1165] [1166] [1167] [1168] [1169] [1170] [1171] [1172] [1173] [1174] [1175] [1176] [1177] [1178] [1179] [1180] [1181] [1182] [1183] [1184] [1185] [1186] [1187] [1188] [1189] [1190] [1191] [1192] [1193] [1194] [1195] [1196] [1197] [1198] [1199] [1200] [1201] [1202] [1203] [1204] [1205] [1206] [1207] [1208] [1209] [1210] [1211] [1212] [1213] [1214] [1215] [1216] [1217] [1218] [1219] [1220] [1221] [1222] [1223] [1224] [1225] [1226] [1227] [1228] [1229] [1230] [1231] [1232] [1233] [1234] [1235] [1236] [1237] [1238] [1239] [1240] [1241] [1242] [1243] [1244] [1245] [1246] [1247] [1248] [1249] [1250] [1251] [1252] [1253] [1254] [1255] [1256] [1257] [1258] [1259] [1260] [1261] [1262] [1263] [1264] [1265] [1266] [1267] [1268] [1269] [1270] [1271] [1272] [1273] [1274] [1275] [1276] [1277] [1278] [1279] [1280] [1281] [1282] [1283] [1284] [1285] [1286] [1287] [1288] [1289] [1290] [1291] [1292] [1293] [1294] [1295] [1296] [1297] [1298] [1299] [1300] [1301] [1302] [1303] [1304] [1305] [1306] [1307] [1308] [1309] [1310] [1311] [1312] [1313] [1314] [1315] [1316] [1317] [1318] [1319] [1320] [1321] [1322] [1323] [1324] [1325] [1326] [1327] [1328] [1329] [1330] [1331] [1332] [1333] [1334] [1335] [1336] [1337] [1338] [1339] [1340] [1341] [1342] [1343] [1344] [1345] [1346] [1347] [1348] [1349] [1350] [1351] [1352] [1353] [1354] [1355] [1356] [1357] [1358] [1359] [1360] [1361] [1362] [1363] [1364] [1365] [1366] [1367] [1368] [1369] [1370] [1371] [1372] [1373] [1374] [1375] [1376] [1377] [1378] [1379] [1380] [1381] [1382] [1383] [1384] [1385] [1386] [1387] [1388] [1389] [1390] [1391] [1392] [1393] [1394] [1395] [1396] [1397] [1398] [1399] [1400] [1401] [1402] [1403] [1404] [1405] [1406] [1407] [1408] [1409] [1410] [1411] [1412] [1413] [1414] [1415] [1416] [1417] [1418] [1419] [1420] [1421] [1422] [1423] [1424] [1425] [1426] [1427] [1428] [1429] [1430] [1431] [1432] [1433] [1434] [1435] [1436] [1437] [1438] [1439] [1440] [1441] [1442] [1443] [1444] [1445] [1446] [1447] [1448] [1449] [1450] [1451] [1452] [1453] [1454] [1455] [1456] [1457] [1458] [1459] [1460] [1461] [1462] [1463] [1464] [1465] [1466] [1467] [1468] [1469] [1470] [1471] [1472] [1473] [1474] [1475] [1476] [1477] [1478] [1479] [1480] [1481] [1482] [1483] [1484] [1485] [1486] [1487] [1488] [1489] [1490] [1491] [1492] [1493] [1494] [1495] [1496] [1497] [1498] [1499] [1500] [1501] [1502] [1503] [1504] [1505] [1506] [1507] [1508] [1509] [1510] [1511] [1512] [1513] [1514] [1515] [1516] [1517] [1518] [1519] [1520] [1521] [1522] [1523] [1524] [1525] [1526] [1527] [1528] [1529] [1530] [1531] [1532] [1533] [1534] [1535] [1536] [1537] [1538] [1539] [1540] [1541] [1542] [1543] [1544] [1545] [1546] [1547] [1548] [1549] [1550] [1551] [1552] [1553] [1554] [1555] [1556] [1557] [1558] [1559] [1560] [1561] [1562] [1563] [1564] [1565] [1566] [1567] [1568] [1569] [1570] [1571] [1572] [1573] [1574] [1575] [1576] [1577] [1578] [1579] [1580] [1581] [1582] [1583] [1584] [1585] [1586] [1587] [1588] [1589] [1590] [1591] [1592] [1593] [1594] [1595] [1596] [1597] [1598] [1599] [1600] [1601] [1602] [1603] [1604] [1605] [1606] [1607] [1608] [1609] [1610] [1611] [1612] [1613] [1614] [1615] [1616] [1617] [1618] [1619] [1620] [1621] [1622] [1623] [1624] [1625] [1626] [1627] [1628] [1629] [1630] [1631] [1632] [1633] [1634] [1635] [1636] [1637] [1638] [1639] [1640] [1641] [1642] [1643] [1644] [1645] [1646] [1647] [1648] [1649] [1650] [1651] [1652] [1653] [1654] [1655] [1656] [1657] [1658] [1659] [1660] [1661] [1662] [1663] [1664] [1665] [1666] [1667] [1668] [1669] [1670] [1671] [1672] [1673] [1674] [1675] [1676] [1677] [1678] [1679] [1680] [1681] [1682] [1683] [1684] [1685] [1686] [1687] [1688] [1689] [1690] [1691] [1692] [1693] [1694] [1695] [1696] [1697] [1698] [1699] [1700] [1701] [1702] [1703] [1704] [1705] [1706] [1707] [1708] [1709] [1710] [1711] [1712] [1713] [1714] [1715] [1716] [1717] [1718] [1719] [1720] [1721] [1722] [1723] [1724] [1725] [1726] [1727] [1728] [1729] [1730] [1731] [1732] [1733] [1734] [1735] [1736] [1737] [1738] [1739] [1740] [1741] [1742] [1743] [1744] [1745] [1746] [1747] [1748] [1749] [1750] [1751] [1752] [1753] [1754] [1755] [1756] [1757] [1758] [1759] [1760] [1761] [1762] [1763] [1764] [1765] [1766] [1767] [1768] [1769] [1770] [1771] [1772] [1773] [1774] [1775] [1776] [1777] [1778] [1779] [1780] [1781] [1782] [1783] [1784] [1785] [1786] [1787] [1788] [1789] [1790] [1791] [1792] [1793] [1794] [1795] [1796] [1797] [1798] [1799] [1800] [1801] [1802] [1803] [1804] [1805] [1806] [1807] [1808] [1809] [1810] [1811] [1812] [1813] [1814] [1815] [1816] [1817] [1818] [1819] [1820] [1821] [1822] [1823] [1824] [1825] [1826] [1827] [1828] [1829] [1830] [1831] [1832] [1833] [1834] [1835] [1836] [1837] [1838] [1839] [1840] [1841] [1842] [1843] [1844] [1845] [1846] [1847] [1848] [1849] [1850] [1851] [1852] [1853] [1854] [1855] [1856] [1857] [1858] [1859] [1860] [1861] [1862] [1863] [1864] [1865] [1866] [1867] [1868] [1869] [1870] [1871] [1872] [1873] [1874] [1875] [1876] [1877] [1878] [1879] [1880] [1881] [1882] [1883] [1884] [1885] [1886] [1887] [1888] [1889] [1890] [1891] [1892] [1893] [1894] [1895] [1896] [1897] [1898] [1899] [1900] [1901] [1902] [1903] [1904] [1905] [1906] [1907] [1908] [1909] [1910] [1911] [1912] [1913] [1914] [1915] [1916] [1917] [1918] [1919] [1920] [1921] [1922] [1923] [1924] [1925] [1926] [1927] [1928] [1929] [1930] [1931] [1932] [1933] [1934] [1935] [1936] [1937] [1938] [1939] [1940] [1941] [1942] [1943] [1944] [1945] [1946] [1947] [1948] [1949] [1950] [1951] [1952] [1953] [1954] [1955] [1956] [1957] [1958] [1959] [1960] [1961] [1962] [1963] [1964] [1965] [1966] [1967] [1968] [1969] [1970] [1971] [1972] [1973] [1974] [1975] [1976] [1977] [1978] [1979] [1980] [1981] [1982] [1983] [1984] [1985] [1986] [1987] [1988] [1989] [1990] [1991] [1992] [1993] [1994] [1995] [1996] [1997] [1998] [1999] [2000] [2001] [2002] [2003] [2004] [2005] [2006] [2007] [2008] [2009] [2010] [2011] [2012] [2013] [2014] [2015] [2016] [2017] [2018] [2019] [2020] [2021] [2022] [2023] [2024] [2025] [2026] [2027] [2028] [2029] [2030] [2031] [2032] [2033] [2034] [2035] [2036] [2037] [2038] [2039] [2040] [2041] [2042] [2043] [2044] [2045] [2046] [2047] [2048] [2049] [2050] [2051] [2052] [2053] [2054] [2055] [2056] [2057] [2058] [2059] [2060] [2061] [2062] [2063
```

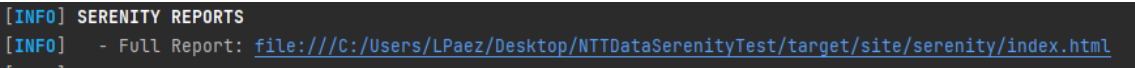
Si la prueba fue exitosa dentro de nuestro proyecto nos generara la siguiente carpeta.



Esta contiene el site, adentro el serenity y dentro de este se encontrará todos en relación a nuestro serenity report. Se valida que exista un archivo index.html que es el que se va abrir para ver el reporte de serenity.



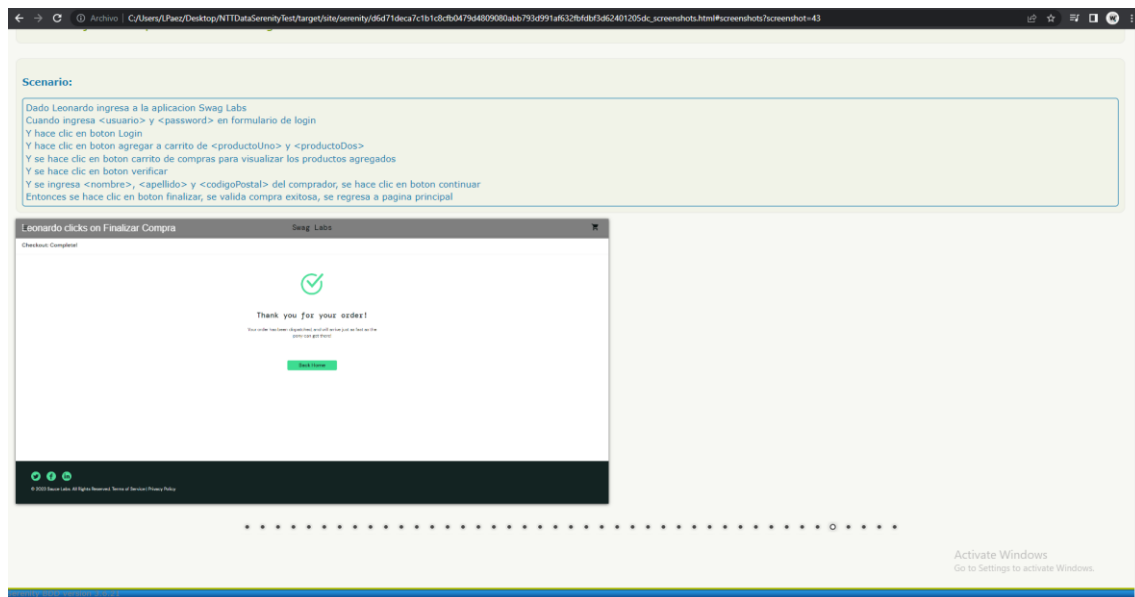
O simplemente para evitar todos estos pasos desde la ejecución que realizamos en el IDE entramos a la ruta que nos proporciona la terminal, la cual apunta al mismo index.html ya mencionado.



Entramos al index.html y validamos el reporte generado, donde podemos ver el porcentaje de éxito y las capturas del paso a paso de nuestra prueba. Esta configuración de capturas se realizo en el serenity properties.



Scenario Outline						
Dado Leonardo ingresa a la aplicacion Swag Labs Cuando ingresa <usuario> y <password> en formulario de login Y hace clic en boton Login Y hace clic en boton agregar a carrito de <productoUno> y <productoDos> Y se hace clic en boton carrito de compras para visualizar los productos agregados Y se hace clic en boton verificar Y se ingresa <nombre>, <apellido> y <codigoPostal> del comprador, se hace clic en boton continuar Entonces se hace clic en boton finalizar, se valida compra exitosa, se regresa a pagina principal						
Examples:						
#	usuario	Password	Producto Uno	Producto Dos	Nombre	Apellido
1	"standard_user"	"secret_sauce"	"Sauce Labs Backpack"	"Test.allTheThings() T-Shirt (Red)"	"Leonardo"	"Paez"
Steps						
1: Flujo de compras dentro de Swag Labs ((usuario="standard_user", password="secret_sauce", productoUno="Sauce Labs Backpack", productoDos="Test.allTheThings() T-Shirt (Red)", nombre="Leonardo", apellido="Paez", codigoPostal="171168"))					Screenshots	Outcome
						SUCCESS 6s 052ms
						SUCCESS 6.28s



Conclusiones

- En conclusión, el caso de prueba fue exitoso. Este queda evidenciado en el serenity report que se va adjuntar como parte los entregables. La versión 3.6.21 del serenity trae consigo varias mejoras a nivel de interfaz en sus reportes. Brindando mayor información del proceso de ejecución por medio de diagramas y tablas de datos.
- Como conclusión o recomendación adicional, en caso de tener varias líneas repetidas dentro de los escenarios de prueba se podría optar por implementar un Background dentro del feature. Esto nos permitiría reutilizar las líneas repetidas en cada uno de los escenarios. Siendo más entendible y más viable de dar mantenibilidad.
- En comparación a versiones antiguas de serenity y de la implementación de la dependencia del webdriver, las versiones actuales reducen esfuerzo a la hora de actualizar el driver para los diferentes navegadores. Únicamente se depende de la configuración de una clase serenity.conf con el driver según su navegador para que realice la descarga automática. Ya no de una descarga y actualización manual como antes se hacía.