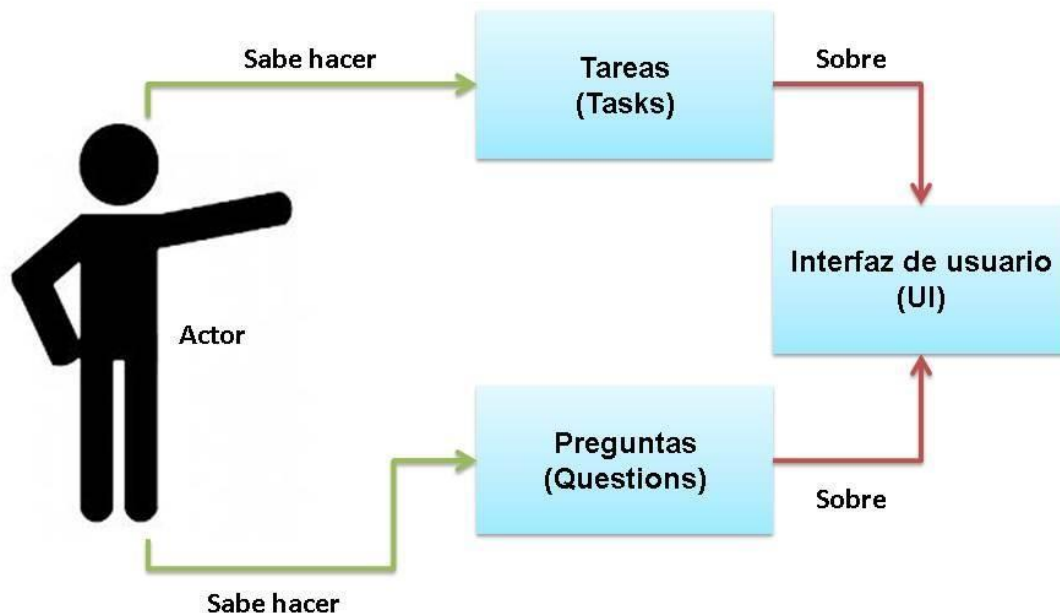


3 SERENITY BDD + SCREENPLAY con CUCUMBER

Primeros pasos con el patrón **Screenplay**

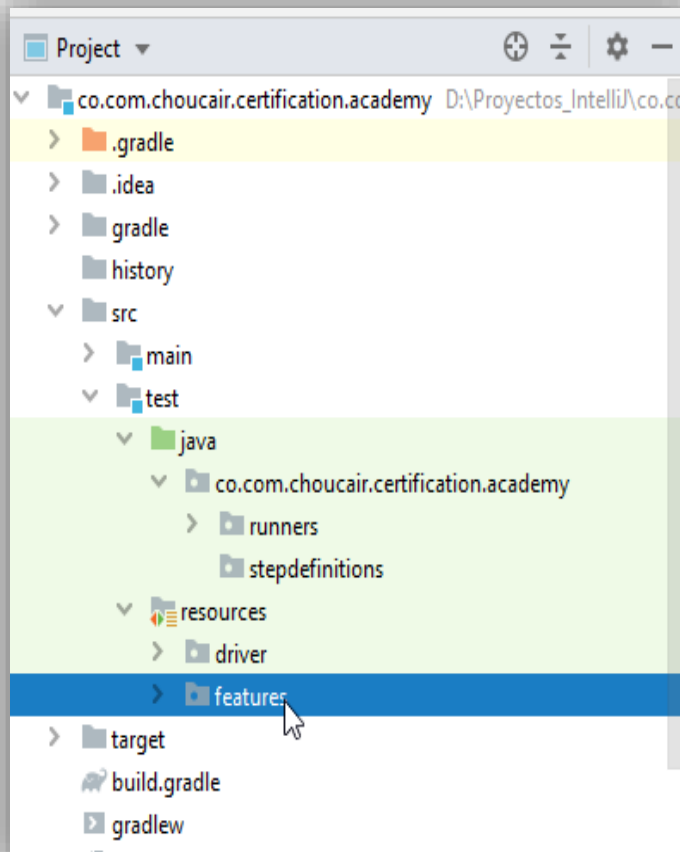


En la presente guía, daremos nuestros primeros pasos sobre el patrón Screenplay. Una vez cargado nuestro proyecto base en eclipse, procederemos a crear una historia de usuario.



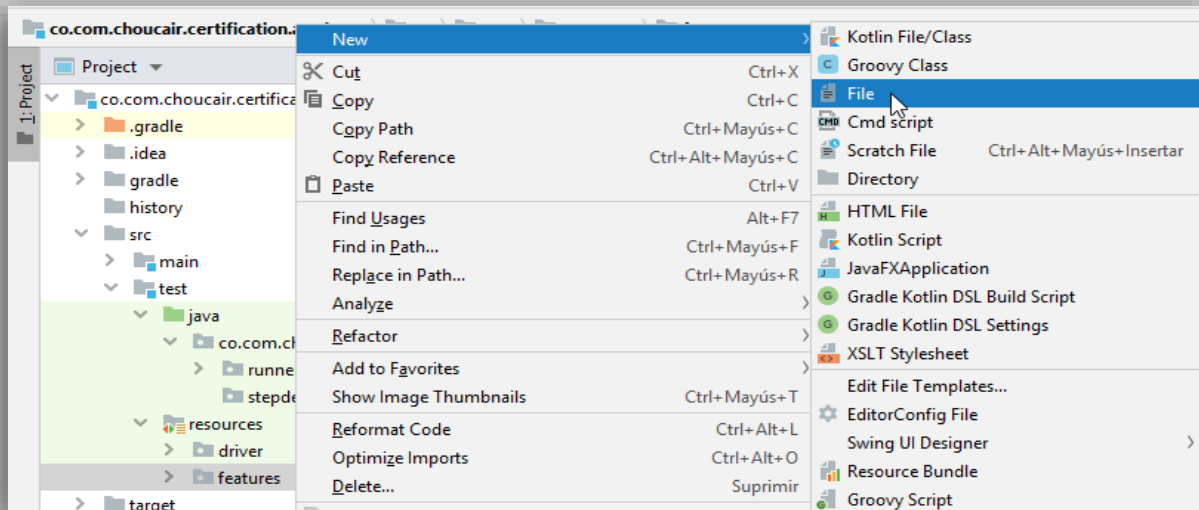
1. Crear una historia de usuario “.feature”

Para esto, iremos a nuestro proyecto y en la carpeta “src/test/resources” encontraremos un paquete llamado “features”.

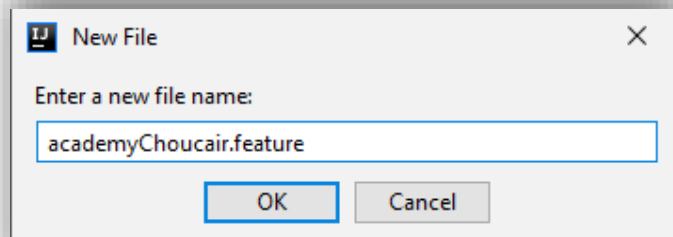


Una vez ahí, presionamos clic derecho sobre el paquete “features”, vamos a “New”>”File”.



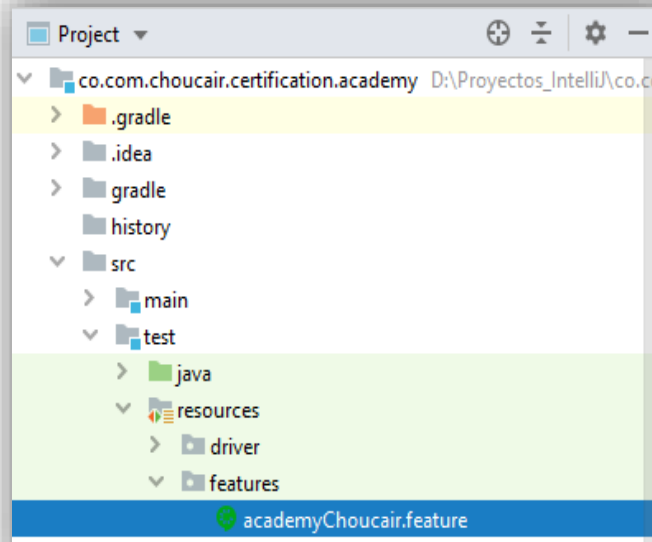


Al escoger la opción File, se presentará una nueva ventana para crear un archivo “.feature”, para esta guía, crearemos uno llamado “academyChoucair”. **Recuerda** que debes añadirle la extensión “.feature” al final.

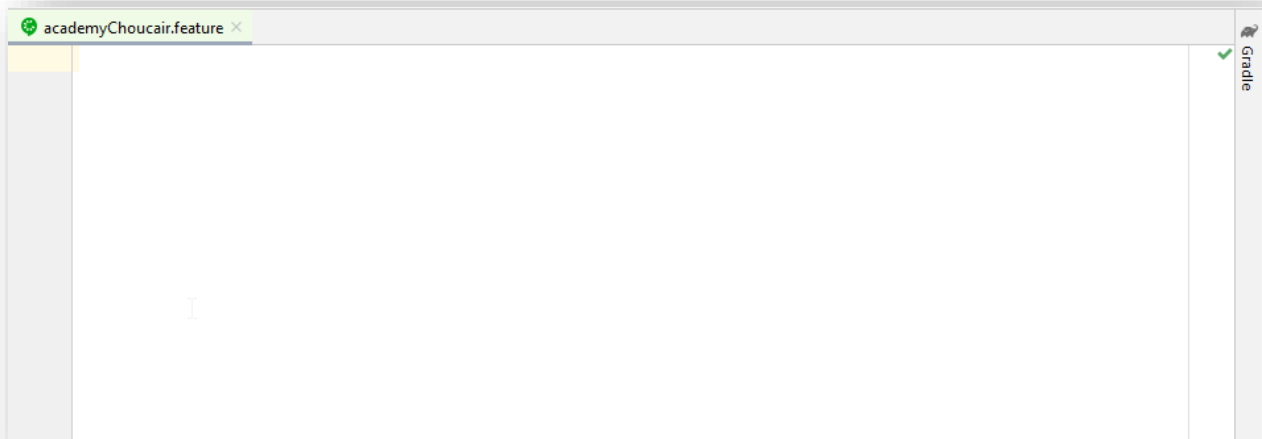


Por último, damos clic en el botón “OK” y verificamos en nuestro proyecto la creación de nuestra feature.





Si tenemos el editor de Cucumber correctamente instalado, al abrir nuestra feature lucirá como la imagen siguiente.



Para esta guía, crearemos una historia de usuario que describa a una persona que abre un navegador, entra a la página de Choucair Academy, escoge de categorías Universidad Choucair, busca cursos “Recursos Automatización Bancolombia” y verifica que el título de la pagina sea “Recursos Automatización Bancolombia”.



2. Escribir la historia de usuario

La idea es que nuestra Feature se escriba en un único idioma, es decir, toda en Inglés o en español, no debemos mezclar distintitos idiomas en la misma.

Para que sea reconocido un idioma diferente al Ingles deberíamos colocar en los Tag iniciales lo siguiente:

```
2 # language:es
```

Con esto estamos determinando que el idioma de nuestra feature es Español.

Gherkin Dialects										
Language	Feature	Background	Scenario	Scenario Outline	Examples	Given	When	Then	And	But
Spanish (es)	Característica	Antecedentes	Escenario	Esquema del escenario	Ejemplos	*	*	*	*	*
						Dado	Cuando	Entonces	Y	Pero
						Dada			E	
						Dados				
						Dadas				

Para nuestro ejemplo realizaremos nuestra **.feature** en inglés, debido a las buenas prácticas “es lo más recomendado”.

Al iniciar con la etiqueta “#Autor:” sirve para dejar como comentario la persona que realiza la feature, aunque esa parte no sea obligatoria.

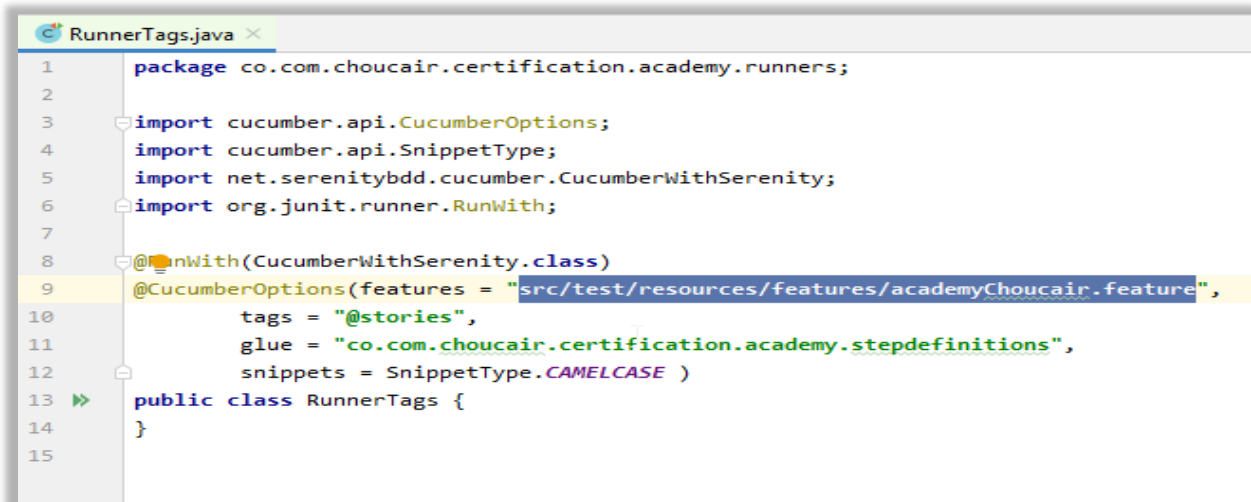
Feature en ingles

```
academyChoucair.feature x
1 # Autor: xxxxxxxxx
2 @stories
3 Feature: Academy Choucair
4   As a user, I want to learn how to automate in screampalay at the Choucair Academy with the automation course
5   @scenario1
6   Scenario: Search for a automation course
7     Given than brandon wants to learn automation at the academy Choucair
8     When he search for the course Recursos Automatización Bancolombia on the choucair academy platform
9     Then he finds the course called resources Recursos Automatización Bancolombia
```

Notamos, que a diferencia del patrón “Page Object Model(POM)”, ahora nuestras acciones las ejecutará un **actor**, en este ejemplo “Brandon” es nuestro actor, y es el que ejecuta todas las acciones de esta prueba. En tu ejemplo, puedes ponerle el nombre que estimes conveniente al actor.

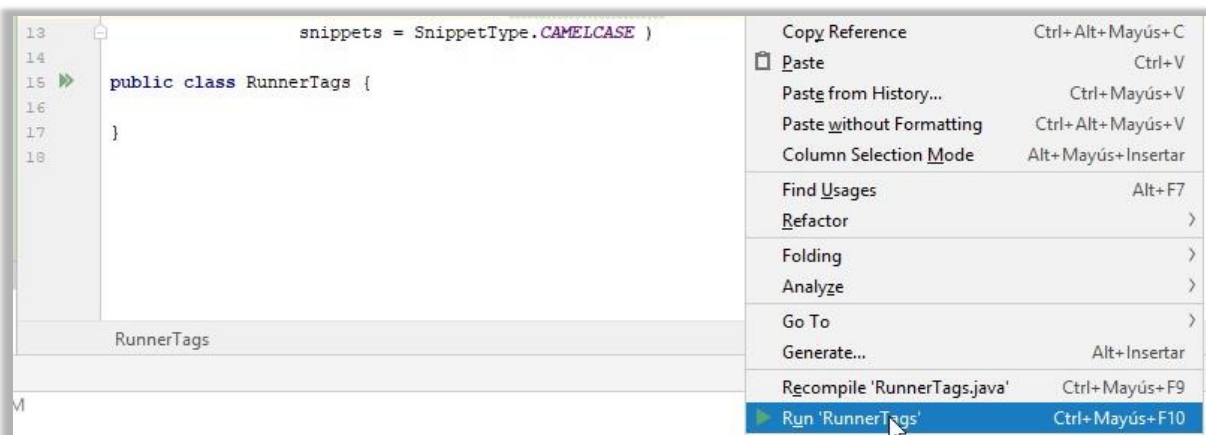
3. Generar los métodos que deben ser agregados a la StepDefinition

Para generar los métodos que se requieren colocar en nuestra clase “StepDefinition”, parametrizamos nuestra clase “RunnerTags” la cual se encuentra en la ruta “**src/test/resources/features/academyChoucair.feature**”, colocando en la misma el nombre de la feature que vamos a ejecutar.



```
1 package co.com.choucair.certification.academy.runners;
2
3 import cucumber.api.CucumberOptions;
4 import cucumber.api.SnippetType;
5 import net.serenitybdd.cucumber.CucumberWithSerenity;
6 import org.junit.runner.RunWith;
7
8 @RunWith(CucumberWithSerenity.class)
9 @CucumberOptions(features = "src/test/resources/features/academyChoucair.feature",
10     tags = "@stories",
11     glue = "co.com.choucair.certification.academy.stepdefinitions",
12     snippets = SnippetType.CAMELCASE )
13 public class RunnerTags {
14 }
15
```

Le daremos clic derecho sobre ésta, y en la opción “Run> RunnerTags”.



Veremos que en la consola nos generará unos métodos sugeridos para copiar y pegar en nuestra clase **“StepDefinition”**

```
Tests ignored: 1 of 1 test – 70 ms

1 Scenarios (1 undefined)
3 Steps (3 undefined)
0m0,958s

You can implement missing steps with the snippets below:

@Given("^than brandon wants to learn automation at the academy Choucair$")
public void thanBrandonWantsToLearnAutomationAtTheAcademyChoucair() throws Exception {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

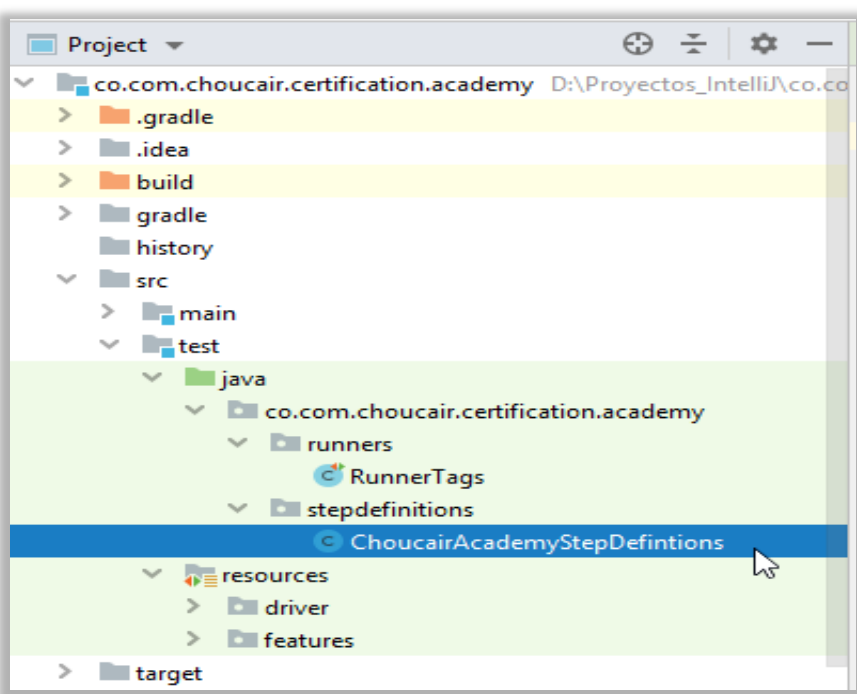
@When("^he search for the course Recursos Automatización Bancolombia on the choucair academy platform$")
public void heSearchForTheCourseRecursosAutomatizaciónBancolombiaOnTheChoucairAcademyPlatform() throws Exception {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}

@Then("^he finds the course called resources Recursos Automatización Bancolombia$")
public void heFindsTheCourseCalledResourcesRecursosAutomatizaciónBancolombia() throws Exception {
    // Write code here that turns the phrase above into concrete actions
    throw new PendingException();
}
```



4. Crear clase StepDefinition

Una vez se hayan generado los métodos sugeridos, nos dirigimos a la ruta del proyecto “src/test/java/co.com.choucair.certification.academy” y en el paquete “**stepdefinitions**” crearemos una clase llamada “ChoucairAcademyStepDefintions”.



Finalmente, pegamos en esta clase los métodos sugeridos anteriormente.




```
ChoucairAcademyStepDefintions.java x
1 package co.com.choucair.certification.academy.stepdefinitions;
2
3 public class ChoucairAcademyStepDefintions {
4
5
6     @Given("^than brandon wants to learn automation at the academy Choucair$")
7     public void thanBrandonWantsToLearnAutomationAtTheAcademyChoucair() throws Exception {
8         // Write code here that turns the phrase above into concrete actions
9         throw new PendingException();
10    }
11
12
13     @When("^he search for the course Recursos Automatización Bancolombia on the choucair academy platform$")
14     public void heSearchForTheCourseRecursosAutomatizaciónBancolombiaOnTheChoucairAcademyPlatform() throws Exception {
15         // Write code here that turns the phrase above into concrete actions
16         throw new PendingException();
17    }
18
19
20     @Then("^he finds the course called resources Recursos Automatización Bancolombia$")
21     public void heFindsTheCourseCalledResourcesRecursosAutomatizaciónBancolombia() throws Exception {
22         // Write code here that turns the phrase above into concrete actions
23         throw new PendingException();
24    }
25 }
```

Importamos las librerías para las annotations (@Given, @When y @Then) o (@Dado, @Cuando y @Entonces).

```
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;
import cucumber.api.java.en.When;
```



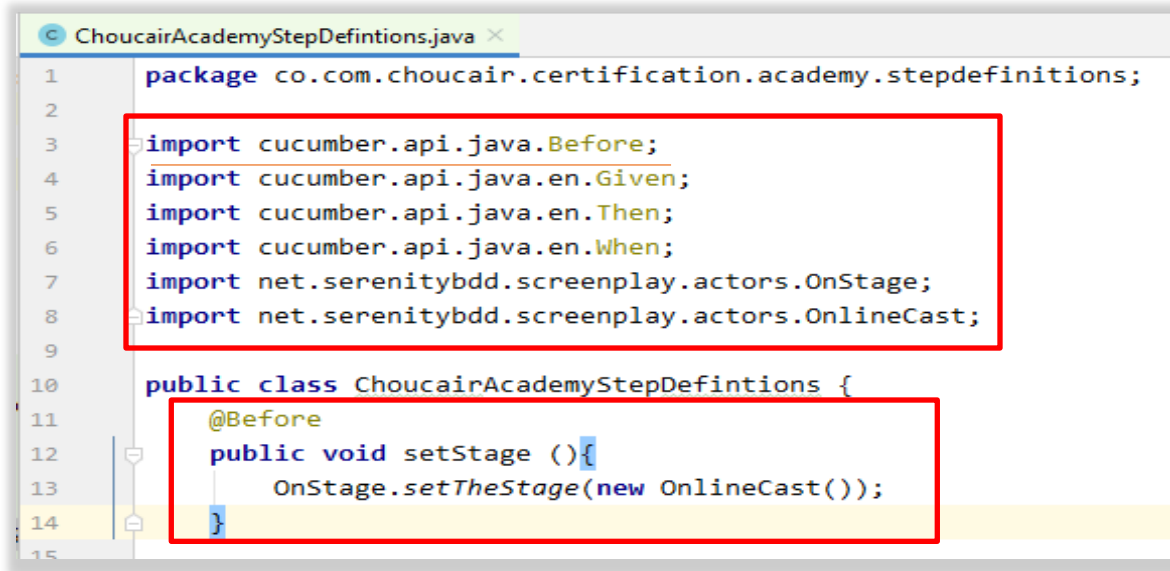
Y por último eliminamos el código sugerido dentro de los métodos. Como resultado, tendremos una clase, que lucirá como la siguiente imagen.

```
ChoucairAcademyStepDefintions.java x
1  package co.com.choucair.certification.academy.stepdefinitions;
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 ChoucairAcademyStepDefintions {
8
9
10     @Given("^than brandon wants to learn automation at the academy Choucair$")
11     public void thanBrandonWantsToLearnAutomationAtTheAcademyChoucair() {
12
13     }
14
15
16     @When("^he search for the course Recursos Automatización Bancolombia on the choucair academy platform$")
17     public void heSearchForTheCourseRecursosAutomatizaciónBancolombiaOnTheChoucairAcademyPlatform() {
18
19     }
20
21
22     @Then("^he finds the course called resources Recursos Automatización Bancolombia$")
23     public void heFindsTheCourseCalledResourcesRecursosAutomatizaciónBancolombia() {
24
25     }
26 }
```



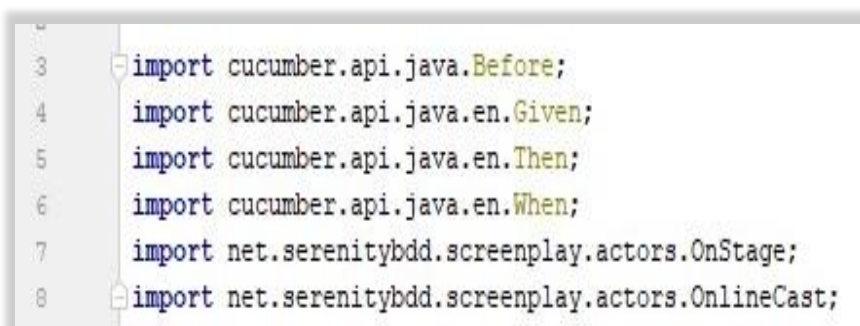
5. Crear Actores y configuración Inicial

Al inicio de nuestra clase “ChoucairAcademyStepDefintions” agregaremos las siguientes tres líneas de código, las cuales se encargan de preparar el Scenario a nuestro Actor:



```
1 package co.com.choucair.certification.academy.stepdefinitions;
2
3 import cucumber.api.java.Before;
4 import cucumber.api.java.en.Given;
5 import cucumber.api.java.en.Then;
6 import cucumber.api.java.en.When;
7 import net.serenitybdd.screenplay.actors.OnStage;
8 import net.serenitybdd.screenplay.actors.OnlineCast;
9
10 public class ChoucairAcademyStepDefintions {
11     @Before
12     public void setStage () {
13         OnStage.setTheStage(new OnlineCast());
14     }
15 }
```

Importamos las librerías para la annotation @Before y nuestro escenario.



```
3 import cucumber.api.java.Before;
4 import cucumber.api.java.en.Given;
5 import cucumber.api.java.en.Then;
6 import cucumber.api.java.en.When;
7 import net.serenitybdd.screenplay.actors.OnStage;
8 import net.serenitybdd.screenplay.actors.OnlineCast;
```

Una vez importadas las librerías, nuestra clase, debe estar de la siguiente forma:



```
ChoucairAcademyStepDefintions.java
1 package co.com.choucair.certification.academy.stepdefinitions;
2
3 import cucumber.api.java.Before;
4 import cucumber.api.java.en.Given;
5 import cucumber.api.java.en.Then;
6 import cucumber.api.java.en.When;
7 import net.serenitybdd.screenplay.actors.OnStage;
8 import net.serenitybdd.screenplay.actors.OnlineCast;
9
10 public class ChoucairAcademyStepDefintions {
11
12     @Before
13     public void setStage () { OnStage.setTheStage(new OnlineCast()); }
14
15
16
17
18     @Given("^than brandon wants to learn automation at the academy Choucair$")
19     public void thanBrandonWantsToLearnAutomationAtTheAcademyChoucair() {
20
21     }
22
23
24     @When("^he search for the course Recursos Automatización Bancolombia on the choucair academy platform$")
25     public void heSearchForTheCourseRecursosAutomatizaciónBancolombiaOnTheChoucairAcademyPlatform() {
26
27     }
28
29     @Then("^he finds the course called resources Recursos Automatización Bancolombia$")
30     public void heFindsTheCourseCalledResourcesRecursosAutomatizaciónBancolombia() {
31
32     }
33
34 }
```

¡Felicidades has dado tus primeros pasos en el patrón
Screenplay!

En nuestra próxima guía ¡Crearemos
Nuestra primera tarea!

