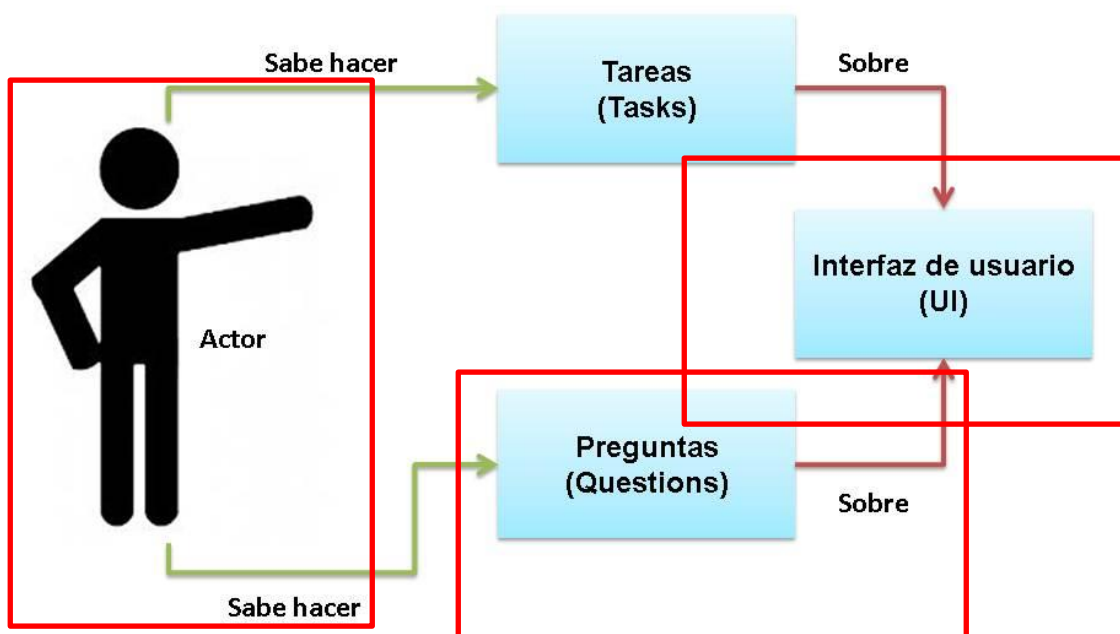


# 6

## SERENITY BDD + SCREENPLAY con CUCUMBER

¡Bienvenidos a nuestra guía 6 del patrón Screenplay!

En esta guía Aprenderemos cómo implementar las verificaciones en el desarrollo de Questions.



Una vez hayamos realizado nuestras tareas anteriores, terminaremos realizando la verificación de que toda nuestra historia de usuario salió conforme a lo esperado. En esta guía buscaremos darle implementación a la siguiente instrucción en nuestra historia de usuario.

**Then** el debería ver la palabra mesa en la pantalla

Entonces iremos nuevamente a nuestra clase “*TraductorGoogleStepDefinition*”, la cual debe lucir de la siguiente forma después de las dos tareas implementadas.

```
1 package co.com.proyectobase.screenplay.stepdefinitions;
2
3+ import org.openqa.selenium.WebDriver;
14
15 public class TraductorGoogleStepDefintions {
16
17-   @Managed(driver="chrome")
18   private WebDriver hisBrowser;
19   private Actor rafa = Actor.named("Rafa");
20
21-   @Before
22   public void configuracionInicial()
23   {
24       rafa.can(BrowseTheWeb.with(hisBrowser));
25   }
26
27-   @Given("^Que Rafa quiere usar el traductor de google$")
28   public void queRafaQuiereUsarElTraductorDeGoogle() {
29       rafa.wasAbleTo(Abrir.LaPaginaDeGoogle());
30   }
31
32-   @When("^el traduce la palabra (.*) del inglés al español$")
33   public void elTraduceLaPalabraTableDelInglésAlEspañol(String palabra) {
34       rafa.attemptsTo(Traducir.DeInglesAEspanol.La(palabra));
35   }
36
37-   @Then("^el debería ver la palabra mesa en la pantalla$")
38   public void elDeberiaVerLaPalabraMesaEnLaPantalla() {
39
40   }
41 }
```

En el método seleccionado dentro del cuadro rojo, es donde estaremos trabajando la implementación para nuestra verificación en el “Question”. Para convertirlo en algo más dinámico, entonces usaremos como en la guía pasada las expresiones regulares pertinentes. Reemplazando la palabra “mesa” por “(.\*)” y declarando un parámetro String, tendremos lo siguiente:



```
1 package co.com.proyectobase.screenplay.stepdefinitions;
2
3+ import org.openqa.selenium.WebDriver;
14
15 public class TraductorGoogleStepDefintions {
16
17-   @Managed(driver="chrome")
18   private WebDriver hisBrowser;
19   private Actor rafa = Actor.named("Rafa");
20
21-   @Before
22   public void configuracionInicial()
23   {
24       rafa.can(BrowseTheWeb.with(hisBrowser));
25   }
26
27-   @Given("^Que Rafa quiere usar el traductor de google$")
28   public void queRafaQuiereUsarElTraductorDeGoogle() {
29       rafa.wasAbleTo(Abrir.LaPaginaDeGoogle());
30   }
31
32-   @When("^el traduce la palabra (.*) del inglés al español$")
33   public void elTraduceLaPalabraTableDelInglésAlEspañol(String palabra) {
34       rafa.attemptsTo(Traducir.DeInglesAEspanolLa(palabra));
35   }
36
37-   @Then("^el deberia ver la palabra (.*) en la pantalla$")
38   public void elDeberiaVerLaPalabraMesaEnLaPantalla(String pregunta) {
39
40   }
41 }
```

Para escribir Questions, usaremos el método “**should**” de nuestro Actor.

```
@Then("^el deberia ver la palabra (.*) en la pantalla$")
public void elDeberiaVerLaPalabraMesaEnLaPantalla(String pregunta) {
    rafa.should();
}
```



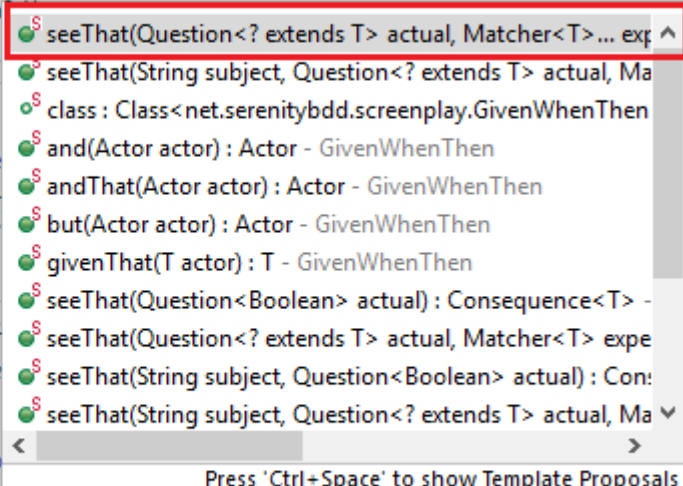
Dentro escribiremos entonces **GivenWhenThen** seguido de un punto y escogeremos el método **seeThat**.

```
@Before
public void configuracionInicia
{
    rafa.can(BrowseTheWeb.with(
})

@Given("^Que Rafa quiere usar e
public void queRafaQuiereUsarEl
    rafa.wasAbleTo(Abrir.LaPagi
})

@When("^el traduce la palabra (
public void elTraduceLaPalabraT
    rafa.attemptsTo(Traducir.De
})

@Then("^el deberia ver la palab
public void elDeberiaVerLaPalab
    rafa.should(GivenWhenThen.
})
```



Obtendremos una línea igual a la siguiente:

```
@Then("^el deberia ver la palabra (.*) en la pantalla$")
public void elDeberiaVerLaPalabraMesaEnLaPantalla(String pregunta) {
    rafa.should(GivenWhenThen.seeThat(actual, expectedMatchers));
}
```

Donde en “actual” escribiremos nuestra pregunta, y en el expectedMatcher escribiremos un **Matcher**, con la información que esperamos obtener. Para escribir la pregunta, seguiremos la misma lógica de una tarea, por lo tanto, obtenemos:

```
@Then("^el deberia ver la palabra (.*) en la pantalla$")
public void elDeberiaVerLaPalabraMesaEnLaPantalla(String pregunta) {
    rafa.should(GivenWhenThen.seeThat(LaRespuesta.es(), expectedMatchers));
}
```

Donde “LaRespuesta” será una clase que crearemos en nuestro paquete “co.com.proyectobase.screenplay.questions”. Y el método “es” será un método estático de dicha clase.

```
rafa.can(BrowseTheWeb.with(hisBrowser));
}

@Given("^Que Rafa quiere usar el traductor de google$")
public void que_Rafa_quiere_usar_el_traductor_de_google() {
    rafa.wasAbleTo(Abrir.LaPaginaDeGoogle());
}

@When("^el traduce la palabra (.*?) del inglés al español$")
public void el_traduce_la_palabra_table_del_ingles_al_espanol(String palabra) {
    rafa.attemptsTo(Traducir.DeInglesAEspanolLa(palabra));
}

@Then("^el deberia ver la palabra (.*?) en la pantalla$")
public void el_deberia_ver_la_palabra_mesa_en_la_pantalla(String palabra) {
    rafa.should(GivenWhenThen.seeThat(LaRespuesta.es( ), Matchers),
}


```

- Matchers - com.google.inject.matcher
- Matchers - org.hamcrest**
- Matchers - org.mockito
- MatchersBinder - org.mockito.internal.invocation
- MatchersPrinter - org.mockito.internal.matchers
- MatchingPageExpressions - net.serenitybdd.core.pages

Press 'Ctrl+Space' to show Template Proposals

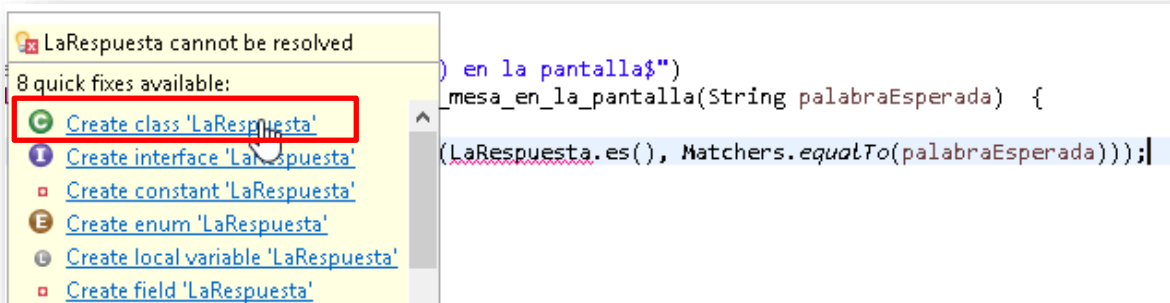
En donde dijimos que iría el “Matcher” de la palabra esperada escribiremos la palabra “**Matchers**”, seguidamente presionamos “Control + barra espaciadora” y escogeremos la clase **Matchers** de **hamcrest**. Luego escribiremos un punto y escogeremos el método “**equalTo()**”. Y escribimos dentro del mismo la palabra que esperamos ver en pantalla.

```
@Then("^el deberia ver la palabra (.*?) en la pantalla$")
public void elDeberiaVerLaPalabraMesaEnLaPantalla(String palabraEsperada) {
    rafa.should(GivenWhenThen.seeThat(LaRespuesta.es(), Matchers.equalTo(palabraEsperada)));
}


```



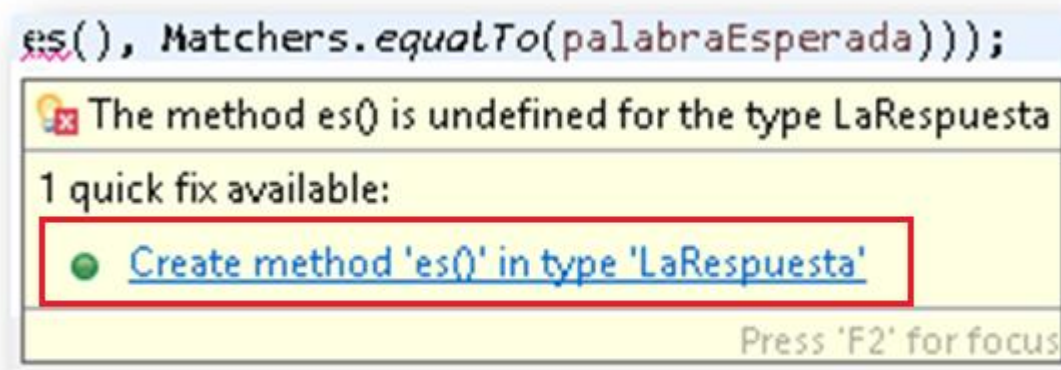
Procedamos a crear nuestra **Question** en el paquete que corresponde:



Verificamos que se haya creado una clase llamada LaRespuesta, y está completamente vacía.

```
1 package co.com.proyectobase.screenplay.questions;  
2  
3 public class LaRespuesta {  
4  
5 }  
6
```

Creamos también el método estático en esta clase:



Y tendremos nuestra clase de la siguiente forma:

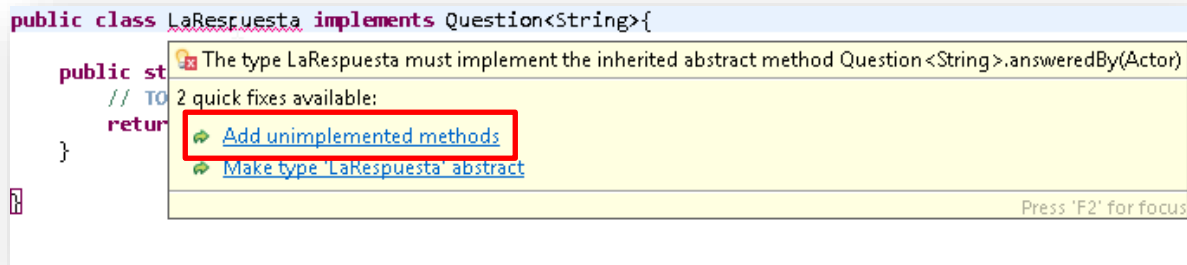
```
1 package co.com.proyectobase.screenplay.questions;
2
3 import net.serenitybdd.screenplay.Question;
4
5 public class LaRespuesta {
6
7     public static Question es() {
8         // TODO Auto-generated method stub
9         return null;
10    }
11
12 }
13
```

Para los Questions usaremos una interface llamada Question y como si fuese del tipo List, le definiremos el tipo que deseamos tener como respuesta, int, String, doublé, etc... En este caso, será del tipo String.

```
1 package co.com.proyectobase.screenplay.questions;
2
3 import net.serenitybdd.screenplay.Question;
4
5 public class LaRespuesta implements Question<String>{
6
7     public static Question es() {
8         // TODO Auto-generated method stub
9         return null;
10    }
11
12 }
13
```



Y agregamos los métodos que no han sido implementados aun:



Nuestra clase debería ir así, después de cambiar el tipo del método “es” al mismo nombre de la clase:

```
1 package co.com.proyectobase.screenplay.questions;  
2  
3 import net.serenitybdd.screenplay.Actor;  
4 import net.serenitybdd.screenplay.Question;  
5  
6 public class LaRespuesta implements Question<String>{  
7  
8     public static LaRespuesta es() {  
9         return null;  
10    }  
11  
12    @Override  
13    public String answeredBy(Actor actor) {  
14  
15        return null;  
16    }  
17  
18 }  
19
```

Agregamos la instanciación de la clase en el método “es” y obtenemos:

```
public static LaRespuesta es() {  
    return new LaRespuesta();  
}
```

Y procedemos a escribir la línea que tomará el texto traducido, desde el elemento del área traducida, inspeccionado en las guías anteriores. En cuanto a las Questions, toda “la magia” la haremos en el método llamado “answeredBy”. Para obtener el String usaremos la sentencia Text.of, le





pasaremos nuestro objeto identificado como un Target, y le diremos que es visto por nuestro actor como un String.

```
8 public class LaRespuesta implements Question<String>{
9
10     public static LaRespuesta es() {
11         return new LaRespuesta();
12     }
13
14     @Override
15     public String answeredBy(Actor actor) {
16
17         return Text.of(GoogleTraductorPage.AREA_TRADUCCION).viewedBy(actor).asString();
18     }
19 }
20 }
21 }
```

Por último para tener un código más “limpio” podemos ir a nuestro TraductorGoogleStepDefinition y hacer unos import static del método seeThat y del equalsTo. Para esto, hagamos lo siguiente. En nuestra clase “**TraductorGoogleStepDefinition**” vamos a la parte superior donde están todas nuestras importaciones y ubicamos el import de nuestro GivenWhenThen, y le añadiremos la palabra static, y al final escribiremos punto y seguidamente escogeremos el nombre del método, en este caso, “seeThat”.

```
import static net.serenitybdd.screenplay.GivenWhenThen.see;
import net.serenitybdd.screenplay.abilities.BrowseTheWeb;
import net.thucydides.core.annotations.Managed;

public class TraductorGoogleStepDefinition {

    @Managed(driver="chrome")
    private WebDriver hisBrowser;
    private Actor rafa = Actor.named("Rafa");

    seeThat(Question<Boolean>) : Consequence<T> - GivenWhenThen
```



Una vez agregada esta línea, podemos ir a la parte inferior, y borrar la palabra `GivenWhenThen`, de la línea de código en nuestro `@Then`.

```
@Then("^el deberia ver la palabra (.*) en la pantalla$")
public void el_deberia_ver_la_palabra_mesa_en_la_pantalla(String palabraEsperada) {
    rafa.should(GivenWhenThen.seeThat LaRespuesta.es(), Matchers.equalTo(palabraEsperada));
}
```

Obtendremos lo siguiente:

```
@Then("^el deberia ver la palabra (.*) en la pantalla$")
public void el_deberia_ver_la_palabra_mesa_en_la_pantalla(String palabraEsperada) {
    rafa.should(seeThat LaRespuesta.es(), Matchers.equalTo(palabraEsperada));
}
```

Haremos lo mismo para nuestro ***Matchers.equalTo()***.

```
import org.hamcrest.Matchers;
import org.openqa.selenium.WebDriver;
```

Y obtendremos al final:

```
3 import static net.serenitybdd.screenplay.GivenWhenThen.seeThat;
4 import static org.hamcrest.Matchers.equalTo;
5
```



Nuestra clase quedaría finalmente así:

```
10
19 public class TraductorGoogleStepDefinition {
20
21     @Managed(driver="chrome")
22     private WebDriver hisBrowser;
23     private Actor rafa = Actor.named("Rafa");
24
25     @Before
26     public void configuracionInicial()
27     {
28         rafa.can(BrowseTheWeb.with(hisBrowser));
29     }
30
31
32     @Given("^Que Rafa quiere usar el traductor de google$")
33     public void que_Rafa_quiere_usar_el_traductor_de_google() {
34         rafa.wasAbleTo(Abrir.LaPaginaDeGoogle());
35     }
36
37     @When("^el traduce la palabra (.*?) del inglés al español$")
38     public void el_traduce_la_palabra_table_del_ingles_al_espanol(String palabra) {
39         rafa.attemptsTo(Traducir.DeInglesAEspanolLa(palabra));
40     }
41
42     @Then("^el deberia ver la palabra (.*?) en la pantalla$")
43     public void el_deberia_ver_la_palabra_mesa_en_la_pantalla(String palabraEsperada) {
44         rafa.should(seeThat(LaRespuesta.es(), equalTo(palabraEsperada)));
45     }
46
47 }
48
49 }
```

Al correr nuestra prueba, y de haber seguido correctamente las guías, veremos que todo ha salido perfecto.

```
-----
[main] INFO net.serenitybdd.core.Serenity - Test Suite Started: TraductorGoogle
[main] INFO net.serenitybdd.core.Serenity -

TEST STARTED

TEST STARTED: Traducir de Ingles a Español
-----
Jun 19, 2018 2:37:12 PM org.openqa.selenium.remote.DesiredCapabilities chrome
INFORMACIÓN: Using 'new ChromeOptions()' is preferred to 'DesiredCapabilities.chrome()'
Starting ChromeDriver 2.39.562718 (9a298c8b0cf5a471a29d30c8b3e12becabb09) on port 13090
Only local connections are allowed.
Jun 19, 2018 2:37:19 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFORMACIÓN: Detected dialect: OSS
[main] INFO net.serenitybdd.core.Serenity -

:) TEST PASSED

TEST PASSED: Traducir de Ingles a Español
-----
```

**¡Felicidades ya conoces la  
implementación básica  
para el patrón Screenplay!  
¡Ahora solo te falta  
ampliar tus conocimientos  
con estudio y práctica!**



