Proyecto Wikipedia completo

1-Feature

consultarEnWikipedia.feature

*#Author: Imendozar@choucairtesting.com*Feature: Wikipedia  
 Como usuario  
 Quiero ingresar a Wikipedia  
 A buscar una palabra  
@case1  
 Scenario: consultar en Wikipedia  
 Given que Luis quiere consultar en Wikipedia  
 When el ingresa la palabra testing en el buscador  
 Then el deberia ver la palabra Historia en la pantalla

2-Runer-

WikipediaRunner

package com.co.choucair.runners;  
  
  
import io.cucumber.junit.CucumberOptions;  
import net.serenitybdd.cucumber.CucumberWithSerenity;  
import org.junit.runner.RunWith;  
  
@RunWith(CucumberWithSerenity.class)  
@CucumberOptions(  
 features = "src/test/resources/features/consultarEnWikipedia.feature",  
 glue = "com.co.choucair.stepdefinitions",  
 tags = " @case1",  
 plugin = {"pretty","json:target/cucumber-reports/cucumber.json"}, //informe en cucumber  
 snippets = CucumberOptions.SnippetType.*CAMELCASE* //formato de steps  
)  
public class WikipediaRunner {  
}  
  
//https://www.youtube.com/watch?v=35iiajZSeJM

3-Step Definitions

WikipediaStepDefinitions

package com.co.choucair.stepdefinitions;  
import static org.hamcrest.Matchers.*equalTo*;  
import com.co.choucair.questions.LaRespuestaQuestions;  
import com.co.choucair.tasks.AbrirWikiTasks;  
import com.co.choucair.tasks.BuscarTasks;  
import io.cucumber.java.Before;  
import io.cucumber.java.en.Given;  
import io.cucumber.java.en.Then;  
import io.cucumber.java.en.When;  
import net.serenitybdd.screenplay.actors.OnlineCast;  
  
import static net.serenitybdd.screenplay.GivenWhenThen.*seeThat*;  
import static net.serenitybdd.screenplay.actors.OnStage.\*;  
  
public class WikipediaStepDefinitions {  
 //Implementación de los metodos Given, When y Then  
//1-prepara nuestra configuración inicial  
 //implementamos un before implementamos un metodo configuracioninicial que setee la escena  
 //onlicast hace parte del actor de serenityy BDD con Screen Play  
 @Before  
 public void configuracionInicial () {*setTheStage*(new OnlineCast());}  
  
  
//2-El actor llamado Luis lo que hace es que abre la pagina wikipedia  
 //el abre la pagina wikipedia llama la clase abrirwikitask  
 @Given("que Luis quiere consultar en Wikipedia")  
 public void queLuisQuiereConsultarEnWikipedia() {  
 *theActorCalled*("luis").wasAbleTo(AbrirWikiTasks.*AbrirLaPaginaDeWikipedia*());  
 }  
  
//3-El actor esta en escena busca la palabra- clase buscar , metodo la, parametro palabra  
 //(.\*) esta notación reemplaza testing y en feature debe aparecer en azulito  
 //parametro palabra de tipo string  
 //el actor esta en el escena busca la palabra ejecuta el metodo la  
 @When("el ingresa la palabra (.\*) en el buscador$")  
 public void elIngresaLaPalabraTestingEnElBuscador(String palabra) {  
 *theActorInTheSpotlight*().attemptsTo(BuscarTasks.*la*(palabra));  
 }  
  
  
  
//el actor en escena lo que hace es que mira la respuesta si la respuesta es  
 // igual a la palabra esperada  
 @Then("el deberia ver la palabra (.\*) en la pantalla$")  
 public void elDeberiaVerLaPalabraHistoriaEnLaPantalla(String palabraEsperada) {  
 *theActorInTheSpotlight*().should((*seeThat*(LaRespuestaQuestions.*es*(),*equalTo*(palabraEsperada))));  
 }  
  
}

4-User interface

a-AbrirWikiUserInterfaces

package com.co.choucair.userinterfaces;  
  
  
import net.serenitybdd.core.annotations.findby.By;  
import net.serenitybdd.core.pages.PageObject;  
import net.serenitybdd.screenplay.targets.Target;  
import net.thucydides.core.annotations.DefaultUrl;  
  
//sse mapean los elementos de la interfaz grafica  
//1-le damos la dirección URL  
@DefaultUrl("https://www.wikipedia.org/")  
  
  
// 2-le decimos que extiende de PageObject  
//El patrón de clase PageObject se usa comúnmente en la automatización de pruebas para representar una página web  
// de una manera orientada a objetos, con métodos que interactúan con los elementos de la página.  
  
public class AbrirWikiUserInterfaces extends PageObject {  
//poner lozalizadores id, xpath del boton o control a usar en pagina web segunda parte buscar  
//nuestro atributo campo de busqueda y boton de busqueda  
 public static final Target *CAMPO\_BUSQUEDA* = Target.*the*("campo de texto de busqueda").located(By.*id*("searchInput"));  
 public static final Target *BOTON\_BUSQUEDA* = Target.*the*("boton de busqueda").located(By.*xpath*("//\*[@id=\"search-form\"]/fieldset/button"));  
  
  
}

b- ResultadoWikiUserInterface

package com.co.choucair.userinterfaces;  
  
import net.serenitybdd.core.annotations.findby.By;  
import net.serenitybdd.screenplay.targets.Target;  
//para verificar que se entro a la página con la palabra va enlazada con Questions  
public class ResultadoWikiUserInterface {  
  
 public static final Target *LABEL\_PALABRA* = Target.*the*("label de palabra resultante").located(By.*id*("Historia"));  
  
}

5-Task

1. AbrirWikiTasks

package com.co.choucair.tasks;  
  
import com.co.choucair.userinterfaces.AbrirWikiUserInterfaces;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.Tasks;  
import net.serenitybdd.screenplay.actions.Open;  
//SOLO PARA ABRIR LA PAGINA WEB ES LA CLASE  
//3-implementamos un task todas sin excepción que es de serenity BDD  
public class AbrirWikiTasks implements Task {  
  
  
 //4-creamos un atributo para abrir la pagina web  
 private AbrirWikiUserInterfaces abrirWikiUserUnterfaces;  
 //5-creamos el metodo de tipo abrir  
 public static AbrirWikiTasks AbrirLaPaginaDeWikipedia () {return Tasks.*instrumented*(AbrirWikiTasks.class);}  
  
 //6-Al metodo sobreescrito le indicamos cual pagina abrira el actor (acción)  
 @Override  
 public <T extends Actor> void performAs(T actor) {actor.attemptsTo(Open.*browserOn*(abrirWikiUserUnterfaces));}  
  
  
  
}

1. BuscarTasks

package com.co.choucair.tasks;  
  
import com.co.choucair.userinterfaces.AbrirWikiUserInterfaces;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.Tasks;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Enter;  
  
  
public class BuscarTasks implements Task {  
  
  
 private String palabra;  
//creamos el metodo constructor y el super  
 public BuscarTasks (String palabra){super (); this.palabra=palabra;}  
// se crea el metodo buscar la  
 public static BuscarTasks la(String palabra) { return Tasks.*instrumented*(BuscarTasks.class, palabra);}  
  
 //le indico las interacciones que tendra con la pagina  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 //acciones que realizara el actor  
 //el actor va ingresar la palabra y va ser un enter dentro del elemento campo busqueda  
 actor.attemptsTo(Enter.*theValue*(palabra).into(AbrirWikiUserInterfaces.*CAMPO\_BUSQUEDA*));  
 //el actor realiza un click en el elemento que esta en AbrirWikiUserInterfaces en elemento boton busqueda  
 actor.attemptsTo(Click.*on*(AbrirWikiUserInterfaces.*BOTON\_BUSQUEDA*));  
 }  
  
  
}

6-question

LaRespuestaQuestions

package com.co.choucair.questions;  
  
import com.co.choucair.userinterfaces.ResultadoWikiUserInterface;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Question;  
import net.serenitybdd.screenplay.questions.Text;  
//aqui va verificar la respuesta  
//implementamos a questions e implementamos su metodo abstracto answeredBy  
public class LaRespuestaQuestions implements Question<String> {  
//creamos em metodo "es" y lo ponemos de tipo LaRespuesta y retornamos un new LaRespuesta  
 public static LaRespuestaQuestions es() {return new LaRespuestaQuestions();}  
  
  
  
 @Override  
 public String answeredBy(Actor actor) {  
 //pasamos el objeto identificado como un target y le decimos que lo vea como String  
 //este validara la respuesta o resultado obtenido  
 //retorne la palabra esperada  
 return Text.*of*(ResultadoWikiUserInterface.*LABEL\_PALABRA*).answeredBy(actor).toString();  
 }  
}