PROYECTO TAMBO

FEATURE

*#Author:* [*Imendozar@choucairtesting.com*](mailto:Imendozar@choucairtesting.com)Feature: Busqueda en pagina web de Tambo  
 Como usuario web  
 Quiero hacer una busqueda en la pagina de Tambo  
 Para poder realizar mi pedido  
  
 @caso2  
 Scenario Outline: Agregar productos en el carrito  
 Given la pagina de web de Tambo  
 When se busca la categoria "<categoria>" y subcategoria "<subcategoria>"  
 And se agregan productos aleatorios al carrito <cantidad>  
  
 Examples:  
 | categoria | subcategoria | cantidad |  
 | Licores | Packs | 5 |  
 | Helados | Personales | 2 |  
 | Cervezas | | 1 |  
  
 @caso3  
 Scenario Outline: Agregar productos en el carrito  
 Given la pagina de web de Tambo  
 When se busca la categoria "<categoria>" y subcategoria "<subcategoria>"  
 And se agregan productos aleatorios al carrito <cantidad>  
 And en el carrito colocar delivery distrito "<distrito>" y direccion "<direccion>"  
 *# Then el carrito mostrar la cantidad*  
 *# And productos ingresados*  
Examples:  
 | categoria | subcategoria | cantidad | distrito | direccion |  
 | Licores | Packs | 1 | Comas | Calle Manco Capac 109 |  
 *# | Helados | Personales | 2 |*  
 *# | Cervezas | | 1 |*

RUNNER

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",  
 glue = "com.co.choucair.stepdefinitions",  
 tags = " @caso3",  
 plugin = {"pretty", "json:target/cucumber-reports/cucumber.json"}, //informe en cucumber  
 snippets = CucumberOptions.SnippetType.*CAMELCASE* //formato de steps  
)  
  
  
public class PedidoTamboRunner {  
}  
  
//https://www.youtube.com/watch?v=MsBlrATU-wU  
//Como automatizar una prueba en menos de 90 min || como utilizar selenium con screenplay

STEP DEFINITION

package com.co.choucair.stepdefinitions;  
  
import com.co.choucair.tasks.CategoriaTasks;  
import com.co.choucair.tasks.ElegirProductosTasks;  
import com.co.choucair.tasks.IngresarDireccionTasks;  
import io.cucumber.java.Before;  
import io.cucumber.java.en.And;  
import io.cucumber.java.en.Given;  
import io.cucumber.java.en.When;  
import net.serenitybdd.screenplay.actions.Open;  
import net.serenitybdd.screenplay.actors.OnStage;  
import net.serenitybdd.screenplay.actors.OnlineCast;  
  
import static net.serenitybdd.screenplay.actors.OnStage.\*;  
public class PedidoTamboStepDefinitions {  
  
 // @Managed(driver = "chrome") //que navegador trabajar  
 // WebDriver driver;  
  
 @Before //inicializa antes que ele Given  
 public void setThestage() {  
 OnStage.*setTheStage*(new OnlineCast()); //como voy apuntar a mis propertis  
 OnStage.*theActorCalled*("Irvin"); //llamar al actor  
 }  
  
 @Given("la pagina de web de Tambo")  
 public void laPaginaDeWebDeTambo() {  
  
 *theActorInTheSpotlight*().wasAbleTo(Open.*url*("https://www.tambo.pe/"));  
 }  
  
 @When("se busca la categoria {string} y subcategoria {string}")  
 public void seBuscaLaCategoriaYSubcategoria(String categoria, String subcategoria) {  
 *theActorInTheSpotlight*().attemptsTo(CategoriaTasks.*metodoCategoria*(categoria,subcategoria));  
 }  
  
  
 @And("se agregan productos aleatorios al carrito {}")  
 public void seAgreganProductosAleatoriosAlCarrito(int cantidad) {  
 OnStage.*theActorInTheSpotlight*().attemptsTo(ElegirProductosTasks.*metodoAgregar*(cantidad));  
 }  
  
 @And("en el carrito colocar delivery distrito {string} y direccion {string}")  
 public void enElCarritoColocarDeliveryDistritoYDireccion(String distrito, String direccion) {  
 *theActorInTheSpotlight*().attemptsTo(IngresarDireccionTasks.*metododireccion*(distrito,direccion));  
 }  
}

USER INTERFACE

package com.co.choucair.userinterfaces;  
  
import net.serenitybdd.screenplay.targets.Target;  
  
public class CategoriaUI {  
  
 public static final Target *CLICK\_MODAL* = Target.*the*("cerrar modal").locatedBy("//div[@role='dialog']/child::button");  
 public static final Target *CLICK\_COOKIS* = Target.*the*("cerrar cookis").locatedBy("//div[@class='text-right space-x-3']/child::button[2]");  
  
 public static final Target *CLICK\_CATEGORIAS* = Target.*the*("click categorias").locatedBy("//header[@id='headerSticky']/descendant::button[1]");  
  
 public static final Target *SELECT\_CATEGORIAS* = Target.*the*("selecciono categorias").locatedBy("//div[@role='group']/child::div//span[text()='{0}']");  
 public static final Target *SELECT\_SUBCATEGORIA* = Target.*the*("selecciono subcategoria").locatedBy("//div[@role='menu']/child::div//span[text()='{0}']");  
  
  
}  
  
  
  
//div[@class='relative product-card']//span[text()='Pack (1 Schweppes Ginger Ale x 1.5 Lt + 1 Ron Cartavio Reserva 8 Años x 750 Ml + Hielo 1.5 Kg)']/ancestor::a/following-sibling::button

package com.co.choucair.userinterfaces;  
  
import net.serenitybdd.screenplay.targets.Target;  
import org.openqa.selenium.By;  
  
public class ElegirProductosUI {  
  
 public static final Target *LISTA\_PRODUCTOS* = Target.*the*("Lista de productos").located(By.*xpath*("//div[contains(@class,'grid gap-5 grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-6')]//div[contains(@class,'relative product-card')]"));  
 public static final Target *BOTON\_AGREGAR\_AL\_CARRITO* = Target.*the*("Botón de agregar al carrito").locatedBy("(//button[contains(@aria-label,'Agregar al carrito')])[{0}]");  
  
 public static final Target *CLICK\_CARRITO* = Target.*the*("click en carrito").locatedBy("//header[@id='headerSticky']/descendant::button[5]");  
  
}

package com.co.choucair.userinterfaces;  
  
import net.serenitybdd.screenplay.targets.Target;  
import org.openqa.selenium.By;  
  
public class IngresarDireccionUI {  
  
 public static final Target *CLICK\_DONDEPEDIR* = Target.*the*("click en donde pedi").locatedBy("//div[@class='pb-5 space-y-2']/child::button/descendant::span");  
  
 public static final Target *IFRAME\_DISTRITO* = Target.*the*("ingresar iframe distrito").locatedBy("//iframe[@title='V3 Address Form']");  
 public static final Target *INPUT\_DISTRITO* = Target.*the*("ingresar distrito").locatedBy("//span[text()='Distrito']/ancestor::label/following-sibling::div[1]//input");  
  
  
 public static final Target *INPUT\_DIRECCION* = Target.*the*("ingresar direccion").locatedBy("//div[@class='flex items-center border-b px-3']/input");  
 public static final Target *BUTTON\_RETIRO* = Target.*the*("seleccionar donde retirar").locatedBy("//div[@role='tablist']/descendant::button[2]");  
  
}

TASK

package com.co.choucair.tasks;  
  
import com.co.choucair.userinterfaces.CategoriaUI;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.questions.WebElementQuestion;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class CategoriaTasks implements Task {  
  
 private final String categoria;  
 private final String subcategoria;  
  
 public CategoriaTasks(String categoria, String subcategoria) {  
 this.categoria = categoria;  
 this.subcategoria = subcategoria;  
 }  
  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 // Cerrar el modal de cookies si está presente y si no esta presente no pasa nada  
 if (WebElementQuestion.*the*(CategoriaUI.*CLICK\_MODAL*).answeredBy(actor).isCurrentlyVisible()) {  
 actor.attemptsTo(Click.*on*(CategoriaUI.*CLICK\_MODAL*));  
 }  
  
 // Cerrar el modal de cookies si está presente y si no esta presente no pasa nada  
 if (WebElementQuestion.*the*(CategoriaUI.*CLICK\_COOKIS*).answeredBy(actor).isCurrentlyVisible()) {  
 actor.attemptsTo(Click.*on*(CategoriaUI.*CLICK\_COOKIS*));  
 }  
  
  
 // Cerrar el modal de cookies si está presente y si no esta presente no pasa nada  
 if (WebElementQuestion.*the*(CategoriaUI.*CLICK\_MODAL*).answeredBy(actor).isCurrentlyVisible()) {  
 actor.attemptsTo(Click.*on*(CategoriaUI.*CLICK\_MODAL*));  
 }  
  
  
 // Esperar hasta que el botón de categorías esté visible  
 actor.attemptsTo(WaitUntil.*the*(CategoriaUI.*CLICK\_CATEGORIAS*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds());  
 // Hacer clic en el botón de categorías  
 actor.attemptsTo(Click.*on*(CategoriaUI.*CLICK\_CATEGORIAS*));  
  
  
  
 // Esperar hasta que la categoría esté visible  
 actor.attemptsTo(WaitUntil.*the*(CategoriaUI.*SELECT\_CATEGORIAS*.of(String.*valueOf*(categoria)), WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds());  
 // Hacer clic en la categoría especificada en la varible categoria  
 actor.attemptsTo(Click.*on*(CategoriaUI.*SELECT\_CATEGORIAS*.of(String.*valueOf*(categoria))));  
  
  
 // Verificar si la subcategoría está presente y hacer clic si lo está  
 if (subcategoria != null && !subcategoria.isEmpty() && WebElementQuestion.*the*(CategoriaUI.*SELECT\_SUBCATEGORIA*.of(String.*valueOf*(subcategoria))).answeredBy(actor).isCurrentlyVisible()) {  
 // Esperar hasta que la subcategoría esté visible  
 actor.attemptsTo(WaitUntil.*the*(CategoriaUI.*SELECT\_SUBCATEGORIA*.of(String.*valueOf*(subcategoria)), WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds());  
 // Hacer clic en la subcategoría especificada en la variable subcategoria  
 actor.attemptsTo(Click.*on*(CategoriaUI.*SELECT\_SUBCATEGORIA*.of(String.*valueOf*(subcategoria))));  
 }  
  
 }  
 public static CategoriaTasks metodoCategoria(String categoria,String subcategoria){  
 return *instrumented*(CategoriaTasks.class,categoria,subcategoria);  
 }  
}

package com.co.choucair.tasks;  
  
import com.co.choucair.userinterfaces.ElegirProductosUI;  
import net.serenitybdd.core.pages.WebElementFacade;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Scroll;  
  
  
import java.util.List;  
import java.util.Random;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class ElegirProductosTasks implements Task {  
  
 private final int cantidad;  
  
 public ElegirProductosTasks(int cantidad) {  
 this.cantidad = cantidad;  
 }  
  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 // Obtener la lista de productos visibles  
 List<WebElementFacade> productos = ElegirProductosUI.*LISTA\_PRODUCTOS*.resolveAllFor(actor);  
  
 // Crear un objeto Random para seleccionar productos aleatorios  
 Random random = new Random();  
  
 for (int i = 0; i < cantidad; i++) {  
 // Seleccionar un producto aleatorio de la lista  
 int indiceAleatorio = random.nextInt(productos.size());  
 WebElementFacade productoAleatorio = productos.get(indiceAleatorio);  
  
 // Esperar hasta que el producto esté visible  
 productoAleatorio.waitUntilVisible();  
  
 // Desplazarse hasta el producto aleatorio  
 actor.attemptsTo(Scroll.*to*(productoAleatorio));  
  
 // Hacer clic en el botón de agregar al carrito del producto aleatorio  
 actor.attemptsTo(Click.*on*(ElegirProductosUI.*BOTON\_AGREGAR\_AL\_CARRITO*.of(String.*valueOf*(indiceAleatorio + 1))));  
 }  
  
 //click al carrito para ver los productos  
 actor.attemptsTo(Click.*on*(ElegirProductosUI.*CLICK\_CARRITO*));  
  
 }  
  
 public static ElegirProductosTasks metodoAgregar(int cantidad) {  
 return *instrumented*(ElegirProductosTasks.class, cantidad);  
 }  
}

package com.co.choucair.tasks;  
  
import com.co.choucair.userinterfaces.IngresarDireccionUI;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.\*;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.openqa.selenium.Keys;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class IngresarDireccionTasks implements Task {  
  
 private final String distrito;  
 private final String direccion;  
  
 public IngresarDireccionTasks(String distrito, String direccion) {  
 this.distrito = distrito;  
 this.direccion = direccion;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 // Hacer clic en el botón "donde pedir"  
 actor.attemptsTo(Click.*on*(IngresarDireccionUI.*CLICK\_DONDEPEDIR*));  
 //actor.attemptsTo(Click.on(IngresarDireccionUI.BUTTON\_RETIRO));  
 // actor.attemptsTo(Click.on(IngresarDireccionUI.INPUT\_DISTRITO));  
  
 // Esperar hasta que el campo distrito esté visible y desplazarse hasta él  
 // Ingresar datos en el campo distrito y presionar Enter  
  
 // Cambiar al iframe  
 actor.attemptsTo(Switch.*toFrame*(IngresarDireccionUI.*IFRAME\_DISTRITO*.resolveFor(actor)));  
  
 actor.attemptsTo(  
 WaitUntil.*the*(IngresarDireccionUI.*INPUT\_DISTRITO*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(IngresarDireccionUI.*INPUT\_DISTRITO*),  
 Enter.*theValue*(distrito).into(IngresarDireccionUI.*INPUT\_DISTRITO*),  
 Hit.*the*(Keys.*ENTER*).into(IngresarDireccionUI.*INPUT\_DISTRITO*)  
 );  
  
  
 // Esperar hasta que el campo dirección esté visible y desplazarse hasta él  
 // Ingresar datos en el campo dirección y presionar Enter  
 actor.attemptsTo(  
 WaitUntil.*the*(IngresarDireccionUI.*INPUT\_DIRECCION*, *isVisible*()).forNoMoreThan(10).seconds(),  
 Scroll.*to*(IngresarDireccionUI.*INPUT\_DIRECCION*),  
 Click.*on*(IngresarDireccionUI.*INPUT\_DIRECCION*),  
 Enter.*theValue*(direccion).into(IngresarDireccionUI.*INPUT\_DIRECCION*),  
 Hit.*the*(Keys.*ENTER*).into(IngresarDireccionUI.*INPUT\_DIRECCION*)  
 );  
  
 }  
 public static IngresarDireccionTasks metododireccion(String distrito,String direccion){  
 return *instrumented*(IngresarDireccionTasks.class,distrito,direccion);  
 }  
}