CAMBIO DE FONDO FINAL

Feature

@CambioFondoAportes  
Feature: Cambio de Fondos  
 Como usuario de la Agencia Digital  
 Quiero cambiar mis fondos de pensión y aportes voluntarios  
 Para gestionar mejor mis inversiones y ahorros  
  
 @Escenario1 @CambioFondo @Happy  
 Scenario Outline: "<Caso de Prueba>" El usuario realiza un cambio de su fondo de pensiones a "<fondo de pensión>" exitoso  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipoDocumento>"  
 And el usuario ingresa el nombre de usuario "<usuario>"  
 And el usuario ingresa la contraseña "<contraseña>"  
 And el usuario hace clic en el botón de inicio de sesión  
 And el usuario cierra va ventana modal informativa  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario hace clic en el botón Cambia tu tipo de fondo  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario elige en tipo de fondo "Tu Fondo de Pensión" el nuevo fondo "<fondo de pensión>"  
 And el usuario elige en tipo de fondo "Tu Aporte Voluntario con Fin Previsional" el nuevo fondo "<aporte voluntario con fin previsional>"  
 And el usuario elige en tipo de fondo "Tu Aporte Voluntario sin Fin Previsional" el nuevo fondo "<aporte voluntario sin fin previsional>"  
 And el usuario hace clic en el botón Continuar  
 Then el usuario debería ver "Este es el resumen de tu solicitud" en la página  
 And el usuario acepta los términos y condiciones  
 And el usuario hace clic en el botón Continuar para enviar código de seguridad  
 And el usuario ingresa a outlook con el correo "[pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe)" y clave "8mbKzo0b7V"  
 And el usuario búsca el último correo con el asunto "AFP Integra - Código de Seguridad para Cambio de Fondo"  
 And el usuario obtiene un código de seguridad  
 And el usuario ingresa el código de seguridad obtenido  
 *# When el usuario hace clic en Confirmar Retiro*  
 *# Then el usuario debería ver el siguiente mensaje de confirmación: "Cambio de fondo exitoso"*  
  
Examples:  
 | Caso de Prueba | tipoDocumento | usuario | contraseña | fondo de pensión | aporte voluntario con fin previsional | aporte voluntario sin fin previsional | correo | clave |  
 | CP01 | CSR | 742402050 | Cercania2024 | Fondo 1 | | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
  
*# pendiente agregar mas casos*  
  
@Escenario2 @SolicitudProceso @UnHappy  
 Scenario Outline: "<Caso de Prueba>" El usuario con tipo de documento "<tipo documento>" valida mensaje que tiene una solicitud de cambio de fondo en proceso  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipoDocumento>"  
 And el usuario ingresa el nombre de usuario "<usuario>"  
 And el usuario ingresa la contraseña "<contraseña>"  
 And el usuario hace clic en el botón de inicio de sesión  
 And el usuario cierra va ventana modal informativa  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario hace clic en el botón Cambia tu tipo de fondo  
 And el usaurio hace clic en el botón Sí, empecemos  
 Then el usuario valida el mensaje "Actualmente no puedes cambiarte de fondo, ya que está en proceso"  
  
 Examples:  
 | Caso de Prueba | tipoDocumento | usuario | contraseña |  
 | CP02 | DNI | 72159283 | Cercania2024 |  
  
 *# usuario mayor de 60años*  
 *# validar en sus datos personales que tenga mayor a 60 (consultar si es mayor o mayor igual)*  
 *# validar que el historial de cambio de fondo esté en estado [Automatico]*  
  
 *# AFILIADO RECIENTE MENOS A 3 MESES*  
 *# EL FONDO SE VISUALIZA EN EL DASHBOARD*  
 *# usuario*  
  
*#escenario alternativo falta implementar (cambio tipo de fondo esta bloqueado en boton si empecemos)*  
@Escenario3 @CambioFondo @Alternativo @Happy  
 Scenario Outline: "<Caso de Prueba>" El usuario realiza un cambio de su fondo de pensiones desde la página principal con exito  
 Given que el usuario está en la página principal  
 And el usuario acepta el cookis  
 And el usuario hace clic en la opción Cambiar tu Tipo de Fondo  
 And el usuario hace clic en el botón iniciar trámite  
 When el usuario selecciona el tipo de documento "<tipoDocumento>"  
 And el usuario ingresa el nombre de usuario "<usuario>"  
 And el usuario ingresa la contraseña "<contraseña>"  
 And el usuario hace clic en el botón de inicio de sesión  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario elige en tipo de fondo "Tu Fondo de Pensión" el nuevo fondo "<fondo de pensión>"  
 And el usuario elige en tipo de fondo "Tu Aporte Voluntario con Fin Previsional" el nuevo fondo "<aporte voluntario con fin previsional>"  
 And el usuario elige en tipo de fondo "Tu Aporte Voluntario sin Fin Previsional" el nuevo fondo "<aporte voluntario sin fin previsional>"  
 And el usuario hace clic en el botón Continuar  
 Then el usuario debería ver "Este es el resumen de tu solicitud" en la página  
 And el usuario acepta los términos y condiciones  
 And el usuario hace clic en el botón Continuar para enviar código de seguridad  
 And el usuario ingresa a outlook con el correo "[pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe)" y clave "8mbKzo0b7V"  
 And el usuario búsca el último correo con el asunto "AFP Integra - Código de Seguridad para Cambio de Fondo"  
 And el usuario obtiene un código de seguridad  
 And el usuario ingresa el código de seguridad obtenido  
 *# When el usuario hace clic en Confirmar Retiro*  
 *# Then el usuario debería ver el siguiente mensaje de confirmación: "Cambio de fondo exitoso"*  
  
Examples:  
 | Caso de Prueba | tipoDocumento | usuario | contraseña | fondo de pensión | aporte voluntario con fin previsional | aporte voluntario sin fin previsional | correo | clave |  
 | CP03 | CSR | 742402050 | Cercania2024 | Fondo 1 | | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 7mbKzo0b7V |

Pasos 1

@Given("que el usuario está en la página de inicio de sesión")  
public void queElUsuarioEstáEnLaPáginaDeInicioDeSesión() {  
 *theActorCalled*("user").wasAbleTo(OpenUrl.*loginAFP*());  
}

Task

package tasks.login;  
  
import net.serenitybdd.annotations.Step;  
import net.serenitybdd.model.environment.ConfiguredEnvironment;  
import net.serenitybdd.model.environment.EnvironmentSpecificConfiguration;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Open;  
import net.thucydides.model.util.EnvironmentVariables;  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class OpenUrl implements Task {  
  
 private final String url;  
 private final EnvironmentVariables environmentVariables;  
  
 public OpenUrl(String url) {  
 this.url = url;  
 this.environmentVariables = ConfiguredEnvironment.*getEnvironmentVariables*();  
 }  
  
 public static Task loginAFP() {  
 String url = "login.url";  
 return *instrumented*(OpenUrl.class, url);  
 }  
  
 @Override  
 @Step("{0} Inicia la página #url")  
 public <T extends Actor> void performAs(T actor) {  
  
 String targetUrl = EnvironmentSpecificConfiguration.*from*(environmentVariables).getProperty(url);  
  
 if (targetUrl == null) {  
 throw new IllegalArgumentException("URL not found in environment variables: " + url);  
 } else {  
 actor.attemptsTo(Open.*url*(targetUrl));  
 }  
 }  
}

PASO 2

@And("el usuario selecciona el tipo de documento {string}")  
public void elUsuarioSeleccionaElTipoDeDocumento(String typeDocument) {  
 *theActorInTheSpotlight*().attemptsTo(SelectDocumentType.*withData*(typeDocument));  
}

package tasks.login;  
  
import net.serenitybdd.annotations.Step;  
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.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.CookiesPage;  
  
public class SelectDocumentType implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SelectDocumentType.class);  
 private final String documentType;  
  
 public SelectDocumentType(String documentType) {  
 this.documentType = documentType;  
 }  
  
 public static SelectDocumentType withData(String documentType) {  
 return new SelectDocumentType(documentType);  
 }  
  
 @Override  
 @Step("{0} selecciona el tipo de documento #documentType")  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 10000; // 10 segundos  
 boolean clicked = false;  
  
 while (System.*currentTimeMillis*() < endTime && !clicked) {  
 try {  
 Thread.*sleep*(500); // Espera inicial de 500 milisegundos  
 if (CookiesPage.*DOCUMENT\_TYPE\_DROPDOWN*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(CookiesPage.*DOCUMENT\_TYPE\_DROPDOWN*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(CookiesPage.*DOCUMENT\_TYPE\_DROPDOWN*)  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El menú desplegable de tipo de documento no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el menú desplegable de tipo de documento. Reintentando...", e);  
 try {  
 Thread.*sleep*(500); // Espera medio segundo antes de reintentar  
 } catch (InterruptedException ie) {  
 Thread.*currentThread*().interrupt();  
 *LOGGER*.error("El hilo fue interrumpido", ie);  
 }  
 }  
 }  
  
 if (!clicked) {  
 *LOGGER*.error("No se pudo hacer clic en el menú desplegable de tipo de documento después de 10 segundos.");  
 return;  
 }  
  
 // Paso 2: Selecciona la opción del DropDown  
 actor.attemptsTo(  
 WaitUntil.*the*(CookiesPage.*CLICK\_OPTION\_TYPE\_DROPDOWN*.of(documentType), WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds(),  
 WaitUntil.*the*(CookiesPage.*CLICK\_OPTION\_TYPE\_DROPDOWN*.of(documentType), WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(CookiesPage.*CLICK\_OPTION\_TYPE\_DROPDOWN*.of(documentType))  
 );  
 }  
}

COOKIS

public static final Target *DOCUMENT\_TYPE\_DROPDOWN* = Target.*the*("Document Type Dropdown")  
 .located(By.*cssSelector*(".Dropdown\_c\_dropdown\_\_relative\_\_zbfiJ"));

public static final Target *CLICK\_OPTION\_TYPE\_DROPDOWN* = Target.*the*("Click drop down option")  
 .locatedBy("//li[@data-keyprop='{0}']");

PASO 3

@And("el usuario ingresa el nombre de usuario {string}")  
public void elUsuarioIngresaElNombreDeUsuario(String user) {  
 *theActorInTheSpotlight*().attemptsTo(InputUser.*withData*(user));  
}

package tasks.login;  
  
import net.serenitybdd.annotations.Step;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Performable;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.Tasks;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.CookiesPage;  
  
public class InputUser implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(InputUser.class);  
 private final String enterUser;  
  
 public InputUser(String enterUser) {  
 this.enterUser = enterUser;  
 }  
  
 public static Performable withData(String enterUser) {  
 return Tasks.*instrumented*(InputUser.class, enterUser);  
 }  
  
 @Override  
 @Step("{0} ingresa usuario")  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 10000; // 10 segundos  
 boolean entered = false;  
  
 while (System.*currentTimeMillis*() < endTime && !entered) {  
 try {  
  
 if (CookiesPage.*USERNAME\_FIELD*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(CookiesPage.*USERNAME\_FIELD*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Enter.*theValue*(enterUser).into(CookiesPage.*USERNAME\_FIELD*)  
 );  
 entered = true;  
 } else {  
 *LOGGER*.info("El campo de entrada de usuario no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de ingresar el usuario. Reintentando...", e);  
 try {  
 Thread.*sleep*(500); // Espera medio segundo antes de reintentar  
 } catch (InterruptedException ie) {  
 Thread.*currentThread*().interrupt();  
 *LOGGER*.error("El hilo fue interrumpido", ie);  
 }  
 }  
 }  
  
 if (!entered) {  
 *LOGGER*.error("No se pudo ingresar el usuario después de 10 segundos.");  
 }  
 }  
}

Cookis

public static final Target *USERNAME\_FIELD* = Target.*the*("Enter Username")  
 .located(org.openqa.selenium.By.*id* ("inpDocumentNumber"));

PASO 4

@And("el usuario ingresa la contraseña {string}")  
public void elUsuarioIngresaLaContraseña(String password) {  
 *theActorInTheSpotlight*().attemptsTo(InputPassword.*withData*(password));  
}

package tasks.login;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.CookiesPage;  
  
public class InputPassword implements Task {  
  
 private final String enterPassword;  
  
 public InputPassword(String enterPassword) {  
 this.enterPassword = enterPassword;  
 }  
  
 public static InputPassword withData(String enterPassword) {  
 return new InputPassword(enterPassword);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 WaitUntil.*the*(CookiesPage.*PASSWORD\_FIELD*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds(),  
 Enter.*theValue*(enterPassword).into(CookiesPage.*PASSWORD\_FIELD*));  
 }  
}

public static final Target *PASSWORD\_FIELD* = Target.*the*("Enter Password")  
 .located(org.openqa.selenium.By.*id*("inpPassword"));

PASO 5

@And("el usuario hace clic en el botón de inicio de sesión")  
public void elUsuarioHaceClicEnElBotónDeInicioDeSesión() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonEnter());  
}

package tasks.login;  
  
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.waits.WaitUntil;  
import ui.CookiesPage;  
  
public class ButtonEnter implements Task {  
  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 WaitUntil.*the*(CookiesPage.*LOGIN\_BUTTON*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(CookiesPage.*LOGIN\_BUTTON*));  
 }  
}

public static final Target *LOGIN\_BUTTON* = Target.*the*("Click on the login button")  
 .located(By.*xpath*("//button[@type='submit']"));

PASO 6

@And("el usuario cierra va ventana modal informativa")  
public void el\_usuario\_cierra\_va\_ventana\_modal\_informativa() {  
 *theActorInTheSpotlight*().attemptsTo(CloseModalPresent.*close*());  
}

package tasks.dashboard;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Scroll;  
import net.serenitybdd.screenplay.conditions.Check;  
import ui.DashboardPage;  
  
public class CloseModalPresent implements Task {  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 Check.*whether*(DashboardPage.*MODAL\_BUTTON*.resolveFor(actor).isVisible())  
 .andIfSo(  
 Scroll.*to*(DashboardPage.*MODAL\_BUTTON*),  
 Click.*on*(DashboardPage.*MODAL\_BUTTON*)  
 )  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static CloseModalPresent close() {  
 return new CloseModalPresent();  
 }  
}

public static final Target *MODAL\_BUTTON* = Target.*the*("Botón del modal")  
 .locatedBy("//div[contains(@class,'IsOpen')]//button");

PASO 7

@Then("el usuario valida el mensaje de bienvenida")  
public void elUsuarioValidaElMensajeDeBienvenida() {  
 *theActorInTheSpotlight*().should(  
 *seeThat*(  
 "El mensaje de bienvenida es correcto",  
 LoginQuestions.*ValidateMessage*(),  
 *anyOf*(  
 *containsString*("Bienvenido a tu Agencia Digital"),  
 *containsString*("Bienvenida a tu Agencia Digital")  
 )  
 )  
 );  
}

Questions

public static Question<String> ValidateMessage() {  
 return actor -> CookiesPage.*MESSAGE\_WELCOME\_LOGIN*.resolveFor(actor).waitUntilVisible().getText().trim();  
}

public static final Target *MESSAGE\_WELCOME\_LOGIN* = Target.*the*("Message Welcome")  
 .locatedBy("//div[@class='i-subtitle']");

PASO 8

@And("el usuario hace clic en el botón Mis trámites del menu")  
public void elUsuarioIngresaElNombreDeUsuario() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonMyProcedures());  
}

package tasks.changeFund;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.DashboardPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class ButtonMyProcedures implements Task {  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 actor.attemptsTo(  
 WaitUntil.*the*(DashboardPage.*MY\_PROCEDURES\_BUTTON*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(DashboardPage.*MY\_PROCEDURES\_BUTTON*));  
 }  
}

Dashboard

public static final Target *MY\_PROCEDURES\_BUTTON* = Target.*the*("Mis trámites")  
 .locatedBy("//a[@href='/mis-tramites']");

PASO 9

@And("el usuario hace clic en el botón Cambia tu tipo de fondo")  
public void elUsuarioHaceClicEnElBotónCambiaTuTipoDeFondo() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonChangeTypeFund());  
}

package tasks.changeFund;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.DashboardPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isClickable*;  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class ButtonChangeTypeFund implements Task {  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 actor.attemptsTo(  
 WaitUntil.*the*(DashboardPage.*CHANGE\_YOUR\_TYPE\_FUND\_BUTTON*, *isVisible*()).forNoMoreThan(30).seconds(),  
 WaitUntil.*the*(DashboardPage.*CHANGE\_YOUR\_TYPE\_FUND\_BUTTON*, *isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(DashboardPage.*CHANGE\_YOUR\_TYPE\_FUND\_BUTTON*)  
 );  
 }  
}

Dasboard

public static final Target *CHANGE\_YOUR\_TYPE\_FUND\_BUTTON* = Target.*the*("Cambia tu tipo de fondo")  
 .locatedBy("//div[text()='Cambia tu Tipo de ']/../..//button[not(@disabled)]");

PASO 10

@And("el usaurio hace clic en el botón Sí, empecemos")  
public void elUsaurioHaceClicEnElBotónSíEmpecemos() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonYesLetStart());  
}

package tasks.changeFund;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.ChangeFundPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class ButtonYesLetStart implements Task {  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 actor.attemptsTo(  
 WaitUntil.*the*(ChangeFundPage.*YES\_LETS\_START\_BUTTON*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ChangeFundPage.*YES\_LETS\_START\_BUTTON*));  
 }  
}

Changefubnd

public static final Target *YES\_LETS\_START\_BUTTON* = Target.*the*("Sí, empecemos")  
 .locatedBy("//div[text()='Sí, empecemos']/..");

PASO 11

@And("el usuario elige en tipo de fondo {string} el nuevo fondo {string}")  
public void elUsuarioEligeEnTipoDeFondoElNuevoFondo(String fund\_type, String fund) {  
 *theActorInTheSpotlight*().attemptsTo(SelectTypeFundAndFund.*withData*(fund\_type, fund));  
}

package tasks.changeFund;  
  
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.targets.Target;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.ChangeFundPage;  
  
public class SelectTypeFundAndFund implements Task {  
  
 private final String typeFund;  
 private final String fund;  
  
 public SelectTypeFundAndFund(String typeFund, String fund) {  
 this.typeFund = typeFund;  
 this.fund = fund;  
 }  
  
 public static SelectTypeFundAndFund withData(String typeFund, String fund) {  
 return new SelectTypeFundAndFund(typeFund,fund);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 if (fund == null || fund.isEmpty()) {  
 return; // No hace nada si el parámetro está vacío  
 }  
  
 Target a = ChangeFundPage.*TYPE\_FUND\_AND\_FUND\_RADIOBUTTON*.of(typeFund, fund);  
 actor.attemptsTo(  
 WaitUntil.*the*(a, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(a));  
 }  
}

Changefund

public static final Target *TYPE\_FUND\_AND\_FUND\_RADIOBUTTON* = Target.*the*("Tipo de fondo {0} y fondo {1}")  
 .locatedBy("//div[text()='{0}']/../..//div[@class='i-text-left']//h4[text()='{1}']");

PASO 12

@And("el usuario elige en tipo de fondo {string} el nuevo fondo {string}")  
public void elUsuarioEligeEnTipoDeFondoElNuevoFondo(String fund\_type, String fund) {  
 *theActorInTheSpotlight*().attemptsTo(SelectTypeFundAndFund.*withData*(fund\_type, fund));  
}

package tasks.changeFund;  
  
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.targets.Target;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.ChangeFundPage;  
  
public class SelectTypeFundAndFund implements Task {  
  
 private final String typeFund;  
 private final String fund;  
  
 public SelectTypeFundAndFund(String typeFund, String fund) {  
 this.typeFund = typeFund;  
 this.fund = fund;  
 }  
  
 public static SelectTypeFundAndFund withData(String typeFund, String fund) {  
 return new SelectTypeFundAndFund(typeFund,fund);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 if (fund == null || fund.isEmpty()) {  
 return; // No hace nada si el parámetro está vacío  
 }  
  
 Target a = ChangeFundPage.*TYPE\_FUND\_AND\_FUND\_RADIOBUTTON*.of(typeFund, fund);  
 actor.attemptsTo(  
 WaitUntil.*the*(a, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(a));  
 }  
}

Changefund

public static final Target *TYPE\_FUND\_AND\_FUND\_RADIOBUTTON* = Target.*the*("Tipo de fondo {0} y fondo {1}")  
 .locatedBy("//div[text()='{0}']/../..//div[@class='i-text-left']//h4[text()='{1}']");

PASO 13

@And("el usuario elige en tipo de fondo {string} el nuevo fondo {string}")  
public void elUsuarioEligeEnTipoDeFondoElNuevoFondo(String fund\_type, String fund) {  
 *theActorInTheSpotlight*().attemptsTo(SelectTypeFundAndFund.*withData*(fund\_type, fund));  
}

package tasks.changeFund;  
  
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.targets.Target;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.ChangeFundPage;  
  
public class SelectTypeFundAndFund implements Task {  
  
 private final String typeFund;  
 private final String fund;  
  
 public SelectTypeFundAndFund(String typeFund, String fund) {  
 this.typeFund = typeFund;  
 this.fund = fund;  
 }  
  
 public static SelectTypeFundAndFund withData(String typeFund, String fund) {  
 return new SelectTypeFundAndFund(typeFund,fund);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 if (fund == null || fund.isEmpty()) {  
 return; // No hace nada si el parámetro está vacío  
 }  
  
 Target a = ChangeFundPage.*TYPE\_FUND\_AND\_FUND\_RADIOBUTTON*.of(typeFund, fund);  
 actor.attemptsTo(  
 WaitUntil.*the*(a, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(a));  
 }  
}

changefund

public static final Target *TYPE\_FUND\_AND\_FUND\_RADIOBUTTON* = Target.*the*("Tipo de fondo {0} y fondo {1}")  
 .locatedBy("//div[text()='{0}']/../..//div[@class='i-text-left']//h4[text()='{1}']");

PASO 14

@And("el usuario hace clic en el botón Continuar")  
public void elUsuarioHafeClicEnElBotónContinuar() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonContinue());  
}

package tasks.changeFund;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Scroll;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.ChangeFundPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class ButtonContinue implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonContinue.class);  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 Scroll.*to*(ChangeFundPage.*CONTINUE\_BUTTON\_1*),  
 WaitUntil.*the*(ChangeFundPage.*CONTINUE\_BUTTON\_1*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ChangeFundPage.*CONTINUE\_BUTTON\_1*)  
 );  
 }  
}

Changefund

public static final Target *CONTINUE\_BUTTON\_1* = Target.*the*("Continuar")  
 .locatedBy("(//button//div[text()='Continuar'])[1]/..");

PASO 15

@Then("el usuario debería ver {string} en la página")  
public void elUsuarioDeberíaVerEnLaPágina(String text) {  
 *theActorInTheSpotlight*().should(*seeThat*(ChangeFundQuestions.*ValidateMessage*(text), *containsString*(text)));  
}

Quetins de changefund

public static Question<String> ValidateMessage(String expectedText) {  
 return actor -> ChangeFundPage.*H5\_LABEL*.of(expectedText).resolveFor(actor).waitUntilVisible().getText().trim();  
}

changefund

public static final Target *H5\_LABEL* = Target.*the*("Válidar el téxto {0}")  
 .locatedBy("//h5[text()='{0}']");

PASO 16

@And("el usuario acepta los términos y condiciones")  
public void elUsuarioAceptaLosTérminosYCondiciones() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(ChangeFundPage.*ACCEPT\_TERMS\_CHECKBOX*, true));  
}

changefund

public static final Target *ACCEPT\_TERMS\_CHECKBOX* = Target.*the*("Aceptar términos y condiciones")  
 .locatedBy("//input[@id='checkAcceptTerms']/..");

PASO 17

@And("el usuario hace clic en el botón Continuar para enviar código de seguridad")  
public void elUsuarioHaceClicEnElBotónContinuarParaEnviarCódigoDeSeguridad() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonContinueToCode());  
}

package tasks.changeFund;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Scroll;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.ChangeFundPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class ButtonContinueToCode implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonContinueToCode.class);  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 Scroll.*to*(ChangeFundPage.*CONTINUE\_BUTTON\_2*),  
 WaitUntil.*the*(ChangeFundPage.*CONTINUE\_BUTTON\_2*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ChangeFundPage.*CONTINUE\_BUTTON\_2*)  
 );  
 }  
}

Changefund

public static final Target *CONTINUE\_BUTTON\_2* = Target.*the*("Continuar")  
 .locatedBy("(//button//div[text()='Continuar'])[2]/..");

PASO 18

@And("el usuario ingresa a outlook con el correo {string} y clave {string}")  
public void elUsuarioIngresaAOutlookConElCorreoYClave(String correo, String clave) {  
  
 String outlookUrl = EnvironmentSpecificConfiguration.*from*(SystemEnvironmentVariables.*createEnvironmentVariables*())  
 .getProperty("environments.default.outlook.url");  
  
 *theActorInTheSpotlight*().attemptsTo(  
 OpenNewTab.*now*(),  
 NavigateToUrl.*to*(outlookUrl),  
 ClickElement.*on*(LoginOutlookPage.*INP\_EMAIL\_USER*,false),  
 EnterText.*into*(LoginOutlookPage.*INP\_EMAIL\_USER*,correo),  
 ClickElement.*on*(LoginOutlookPage.*NEXT\_BUTTON*,false),  
 WaitFor.*seconds*(5),  
 ClickElement.*on*(LoginOutlookPage.*INP\_PASSWORD\_USER*,false),  
 EnterText.*into*(LoginOutlookPage.*INP\_PASSWORD\_USER*,clave),  
 ClickElement.*on*(LoginOutlookPage.*NEXT\_BUTTON*,false),  
 ClickElement.*on*(LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*,false, "Sí"),  
 WaitForPageToLoad.*withTimeout*(10),  
 ClickElement.*on*(OutlookPage.*ACCOUNT\_1*, false),  
 WaitForPageToLoad.*withTimeout*(10)  
 );  
}

Outllok

public static final Target *INP\_EMAIL\_USER* = Target.*the*("Ingresar el usuario del correo")  
 .located(By.*xpath*("//input[@name='loginfmt'] | //input[@type='email']"));

public static final Target *NEXT\_BUTTON* = Target.*the*("next button")  
 .located(By.*xpath*("//\*[@type='submit']"));

public static final Target *INP\_PASSWORD\_USER* = Target.*the*("Enter Password Email Outlook")  
 .locatedBy("//input[@name='passwd']");

public static final Target *NEXT\_BUTTON* = Target.*the*("next button")  
 .located(By.*xpath*("//\*[@type='submit']"));

public static final Target *BTN\_YES\_NOT\_OPTION* = Target.*the*("Click en el botón '{0}' recordar sesión")  
 .locatedBy("//input[@value='{0}']");

public static final Target *ACCOUNT\_1* = Target.*the*("Primera cuenta")  
 .locatedBy("(//div[@role='listitem']//img)[1]");

PASO 19

@And("el usuario búsca el último correo con el asunto {string}")  
public void elUsuarioBúscaElÚltimoCorreoConElAsunto(String asunto) {  
 *theActorInTheSpotlight*().attemptsTo(  
 WaitFor.*seconds*(15),  
 ClickElement.*on*(OutlookPage.*TXT\_SEARCH*,false),  
 EnterText.*into*(OutlookPage.*TXT\_SEARCH*,asunto),  
 ClickElement.*on*(OutlookPage.*BTN\_SEARCH*,false),  
 WaitFor.*seconds*(2),  
 SelectFirstEmail.*now*()  
 );  
}

outllok

public static final Target *TXT\_SEARCH* = Target.*the*("búsqueda")  
 .locatedBy("//input[@id='topSearchInput']");

public static final Target *BTN\_SEARCH* = Target.*the*("botón search")  
 .locatedBy("//button[@title='Buscar']");

PASO 20

@And("el usuario obtiene un código de seguridad")  
public void elUsuarioObtieneUnCódigoDeSeguridad() {  
 *theActorInTheSpotlight*().attemptsTo(SearchSecurityCode.*on*());  
  
}

package tasks.outlook;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Scroll;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
  
import tasks.common.SwitchToTab;  
import tasks.common.WaitFor;  
  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
import static ui.OutlookPage.\*;  
import static ui.OutlookPage.*LBL\_SECURITY\_CODE*;  
  
public class SearchSecurityCode implements Task {  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 actor.attemptsTo(  
 WaitFor.*seconds*(5),  
 WaitUntil.*the*(*WIDE\_CONTENT*, *isVisible*()).forNoMoreThan(30).seconds(),  
 Scroll.*to*(*WIDE\_CONTENT*)  
 );  
  
 actor.attemptsTo(  
  
 // Esperar hasta que el elemento sea visible  
 WaitFor.*seconds*(5),  
 WaitUntil.*the*(*LBL\_SECURITY\_CODE*, *isVisible*()).forNoMoreThan(30).seconds(),  
 // Desplazarse al elemento LBL\_SECURITY\_CODE  
 Scroll.*to*(*LBL\_SECURITY\_CODE*)  
 );  
  
 //Obtener el codigo  
 String securityCode = *LBL\_SECURITY\_CODE*.resolveFor(actor).getText();  
  
 // Guardar el código de seguridad  
 actor.remember("securityCode", securityCode);  
  
 //Vuelve a la primera pestaña  
 *theActorInTheSpotlight*().attemptsTo(SwitchToTab.*to*(0));  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static SearchSecurityCode on() {  
 return new SearchSecurityCode();  
 }  
}

Outllok

public static final Target *WIDE\_CONTENT* = Target.*the*("Contenido")  
 .locatedBy("//div[@class='wide-content-host']");

public static final Target *LBL\_SECURITY\_CODE* = Target.*the*("security code")  
 .locatedBy("//td[@class='x\_code']//div//p");

PASO 21

@And("el usuario ingresa el código de seguridad obtenido")  
public void elUsuarioIngresaElCódigoDeSeguridadObtenido() {  
 String securityCode = *theActorInTheSpotlight*().recall("securityCode");  
 *theActorInTheSpotlight*().attemptsTo(EnterSecurityCodeInFields.*with*(securityCode));  
}

package tasks.changeFund;  
  
import net.serenitybdd.annotations.Step;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.ChangeFundPage;  
  
public class EnterSecurityCodeInFields implements Task {  
 private final String securityCode;  
  
 public EnterSecurityCodeInFields(String securityCode) {  
 this.securityCode = securityCode;  
 }  
  
 @Override  
 @Step("{0} ingresa el código de seguridad en los campos correspondientes")  
 public <T extends Actor> void performAs(T actor) {  
 // Asegurarse de que el código de seguridad tiene 6 caracteres  
 if (securityCode.length() != 6) {  
 throw new IllegalArgumentException("El código de seguridad debe tener 6 caracteres");  
 }  
  
 // Validar que el primer campo de seguridad exista  
 actor.attemptsTo(  
 WaitUntil.*the*(ChangeFundPage.*SECURITY\_CODE\_FIELD*.of("1"), WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds()  
 );  
  
 // Iterar sobre cada carácter del código y colocarlo en el cuadro de texto correspondiente  
 for (int i = 0; i < 6; i++) {  
 String character = String.*valueOf*(securityCode.charAt(i));  
 actor.attemptsTo(  
 Enter.*theValue*(character).into(ChangeFundPage.*SECURITY\_CODE\_FIELD*.of(String.*valueOf*(i + 1)))  
 );  
 }  
 }  
  
 public static EnterSecurityCodeInFields with(String securityCode) {  
 return new EnterSecurityCodeInFields(securityCode);  
 }  
}

public static final Target *SECURITY\_CODE\_FIELD* = Target.*the*("Código de seguridad nro{0}")  
 .locatedBy("//input[@id='inputtokencode{0}']");

PASO 22 ACTUALIZAR PASTH

@When("el usuario hace clic en Confirmar Retiro")  
public void elUsuarioHaceClicEnConfirmarRetiro() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(WithdrawVoluntaryContributionsPage.*CONFIRM\_WITHDRAW\_BUTTON*, true));  
}

public static final Target *CONFIRM\_WITHDRAW\_BUTTON* = Target.*the*("Confirmar retiro")  
 .locatedBy("//button//div[text()='Confirmar retiro']/..");

PASO 23

FALTA IMPLEMENTAR

*Then el usuario debería ver el siguiente mensaje de confirmación: "Cambio de fondo exitoso"*