TRANFERIR AFP ESC 2

FEATURE

TransferAFP.feature

Scenario Outline: <Caso de Prueba> El usuario valida el mensaje "<mensaje>" al realizar un traspaso AFP Integra  
 Given que el usuario está en la página principal  
 And el usuario acepta el cookis  
 When el usuario hace clic en el botón de Hazte cliente  
 And el usuario hace clic en el botón Cámbiate Ya  
 And el usuario hace clic en el botón Cámbiarme aquí "De la Cabecera"  
 And el usuario ingresa "<DNI>" en DNI  
 And el usuario ingresa "<correo>" en Correo Electrónico  
 And el usuario ingresa "<correo>" en Confirmar Correo Electrónico  
 And el usuario ingresa "<celular>" en Celular  
 And el usuario "marca" el check de mejores oportunidades de ahorro e inversión  
 And el usuario sigue el proceso haciendo clic en el botón Continuar  
 Then el usuario valida el mensaje "<mensaje>" al realizar un traspaso AFP Integra  
  
 Examples:  
 | Caso de Prueba | DNI | correo | celular | mensaje |  
 | CP06 | 45335136 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Tú ya eres parte de AFP Integra |  
 | CP07 | 99999999 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Actualmente no te encuentras en una AFP |  
 | CP08 | 06936146 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Actualmente no te encuentras en una AFP |  
 | CP09 | 75953952 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Actualmente no te encuentras en una AFP |  
 | CP10 | 44545454 | [proveedorirvin.mendoza@sura.pe](mailto:proveedorirvin.mendoza@sura.pe) | 929214450 | Cuéntanos, ¿actualmente vives en la dirección que aparece en tu DNI? |

Step definitions

@And("el usuario acepta el cookis")  
public void elUsuarioAceptaElCookis() {  
 *theActorInTheSpotlight*().attemptsTo(WaitFor.*seconds*(10));  
 *theActorInTheSpotlight*().attemptsTo(Check.*whether*(CookiesPage.*COOKIES\_ACCEPT*.resolveFor(  
 *theActorInTheSpotlight*()).isVisible()).andIfSo(ClickElement.*on*(CookiesPage.*COOKIES\_ACCEPT*, true)));  
}

@And("el usuario ingresa {string} en DNI")  
public void elUsuarioIngresaEnDNI(String DNI) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_DNI*,  
 DNI.trim()));  
}

@And("el usuario ingresa {string} en Correo Electrónico")  
public void elUsuarioIngresaEnCorreoElectrónico(String correo) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_CORREO*,  
 correo.trim()));  
}  
  
@And("el usuario ingresa {string} en Confirmar Correo Electrónico")  
public void elUsuarioIngresaEnConfirmarCorreoElectrónico(String correo) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_CONFIRMAR\_CORREO*,  
 correo.trim()));  
}  
  
@And("el usuario ingresa {string} en Celular")  
public void elUsuarioIngresaEnCelular(String celular) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_CELULAR*,  
 celular.trim()));  
}

@And("el usuario {string} el check de mejores oportunidades de ahorro e inversión")  
public void elUsuarioElCheckDeMejoresOportunidadesDeAhorroEInversión(String marca) {  
 if (marca.trim().equalsIgnoreCase("marca")){  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferFormPage.*CHECK\_INFO*,false));  
 }  
}

@And("el usuario sigue el proceso haciendo clic en el botón Continuar")  
public void elUsuarioSigueElProcesoHaciendoClicEnElBotónContinuar() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferFormPage.*BUTTON\_CONTINUAR*, true));  
  
}

@Then("el usuario valida el mensaje {string} al realizar un traspaso AFP Integra")  
 public void elUsuarioValidaElMensajeAlRealizarUnTraspasoAFPIntegra(String mensaje) {  
 // theActorInTheSpotlight().attemptsTo(ScrollTo.element(TransferFormPage.TEXT\_VALIDAR));  
 *theActorInTheSpotlight*().attemptsTo(WaitFor.*seconds*(15)); //tiempo explicito  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 TransferFormPage.*TEXT\_VALIDAR*, mensaje), *is*(true)));  
 }  
}

Ui

package ui;  
  
import net.serenitybdd.screenplay.targets.Target;  
  
public class TransferFormPage {  
  
 public static final Target *INPUT\_DNI* = Target.*the*("escribir DNI")  
 .locatedBy("//input[@data-testid='inpDocumentNumber']");  
  
 public static final Target *INPUT\_CORREO* = Target.*the*("escribir correo")  
 .locatedBy("//input[@data-testid='inpEmail']");  
  
 public static final Target *INPUT\_CONFIRMAR\_CORREO* = Target.*the*("escribir confirmar correo")  
 .locatedBy("//input[@data-testid='inpRepeatEmail']");  
  
 public static final Target *INPUT\_CELULAR* = Target.*the*("escribir celular")  
 .locatedBy("//input[@data-testid='inpMobilephoneNumber']");  
  
 public static final Target *CHECK\_INFO* = Target.*the*("selecciona el check")  
 .locatedBy("//p[contains(text(),'Para recibir mejores oportunidades de ahorro e inversión, autorizo los ')]/parent::span");  
  
 public static final Target *BUTTON\_CONTINUAR* = Target.*the*("click en el botón continuar")  
 .locatedBy("//button//div[text()='Continuar']/..");  
  
 public static final Target *TEXT\_VALIDAR* = Target.*the*("validar texto al iniciar transferir AFP Integra")  
 .locatedBy("//div[contains(@class,'i-body') and contains(text(),'{0}')]");

TASK COMMUN

package tasks.common;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.targets.Target;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class ClickElementJs implements Task {  
 private final Target target;  
  
 public ClickElementJs(Target target) {  
 this.target = target;  
 }  
  
 public static ClickElementJs on(Target target) {  
 return *instrumented*(ClickElementJs.class, target);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 WebDriver driver = net.serenitybdd.screenplay.abilities.BrowseTheWeb.*as*(actor).getDriver();  
 WebElement element = target.resolveFor(actor);  
  
 // Usar JavaScript para hacer clic en el elemento  
 ((JavascriptExecutor) driver).executeScript("arguments[0].click();", element);  
 }  
}

package tasks.common;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.targets.Target;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class ScrollTo implements Task {  
 private final Target target;  
 public ScrollTo(Target target) {  
 this.target = target;  
 }  
  
 public static ScrollTo element(Target target) {  
 return *instrumented*(ScrollTo.class, target);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 WebDriver driver = net.serenitybdd.screenplay.abilities.BrowseTheWeb.*as*(actor).getDriver();  
 WebElement element = target.resolveFor(actor);  
 // Usar JavaScript para hacer scroll al elemento  
 ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);", element);  
 }  
}

ackage tasks.common;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class WaitFor implements Task {  
 private final int seconds;  
  
 public WaitFor(int seconds) {  
 this.seconds = seconds;  
 }  
  
 public static WaitFor seconds(int seconds) {  
 return *instrumented*(WaitFor.class, seconds);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 try {  
 Thread.*sleep*(seconds \* 1000L); // Convertir segundos a milisegundos  
 } catch (InterruptedException e) {  
 Thread.*currentThread*().interrupt(); // Restaurar el estado de interrupción  
 throw new RuntimeException("La espera fue interrumpida", e);  
 }  
 }  
}

EXTRA

QUESTIONS

package questions;  
  
import net.serenitybdd.screenplay.Question;  
import ui.ChangeFundPage;  
  
public class ChangeFundQuestions {  
  
 public static Question<String> ValidateMessage(String expectedText) {  
 return actor -> ChangeFundPage.*H5\_LABEL*.of(expectedText).resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
  
 public static Question<String> ValidateMessageChangeFund(String expectedText) {  
 return actor -> ChangeFundPage.*LBL\_MESSAGE\_CHANGE\_FUND*.of(expectedText).resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
}

package questions;  
  
import net.serenitybdd.screenplay.Question;  
import ui.CookiesPage;  
import ui.LoginPage;  
  
public class LoginQuestions {  
  
 public static Question<String> ValidateMessage() {  
 return actor -> CookiesPage.*MESSAGE\_WELCOME\_LOGIN*.resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
  
 public static Question<String> ValidateMessageFieldUser() {  
 return actor -> CookiesPage.*FIELD\_INVALID\_MESSAGE\_USER*.resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
  
 public static Question<String> ValidateMessageModalUser() {  
 return actor -> CookiesPage.*FIELD\_INVALID\_MESSAGE\_MODAL\_USER*.resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
  
 public static Question<String> ValidateMessageFieldPassword() {  
 return actor -> CookiesPage.*FIELD\_INVALID\_MESSAGE\_PASSWORD*.resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
  
 public static Question<String> ValidateMessagedos() {  
 return actor -> LoginPage.*MENSAJE\_PASSWORD\_ERROR*.resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
  
}

package questions;  
  
import net.serenitybdd.screenplay.Question;  
import ui.ResetPasswordPage;  
  
public class ResetPasswordQuestions {  
  
 public static Question<String> ValidateMessageConfirmation() {  
 return actor -> *normalizeText*(ResetPasswordPage.*MESSAGE\_EMAIL\_VERIFICATION*.resolveFor(actor).waitUntilVisible().getText().trim());  
 }  
  
 private static String normalizeText(String text) {  
 return text.replaceAll("\\s+", " ").trim();  
 }  
  
 public static Question<String> ValidateMessageConfirmationEmail() {  
 return actor -> ResetPasswordPage.*MESSAGE\_HAPPY\_VERIFICATION\_EMAIL*.resolveFor(actor).waitUntilVisible().getText().trim();  
 }  
}

TASK changeFund

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 ButtonCambiarFondo implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonCambiarFondo.class);  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 Scroll.*to*(ChangeFundPage.*BUTTON\_CAMBIAR\_FONDO*),  
 WaitUntil.*the*(ChangeFundPage.*BUTTON\_CAMBIAR\_FONDO*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ChangeFundPage.*BUTTON\_CAMBIAR\_FONDO*)  
 );  
 }  
}

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*)  
 );  
 }  
}

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*)  
 );  
 }  
}

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*)  
 );  
 }  
}

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*));  
 }  
}

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 ButtonValidateCode implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonValidateCode.class);  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ChangeFundPage.*VALIDATE\_CODE\_BUTTON*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ChangeFundPage.*VALIDATE\_CODE\_BUTTON*)  
 );  
 }  
}

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*));  
 }  
}

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 org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.ChangeFundPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class CheckTermsAndConditions implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(CheckTermsAndConditions.class);  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ChangeFundPage.*ACCEPT\_TERMS\_CHECKBOX*, *isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ChangeFundPage.*ACCEPT\_TERMS\_CHECKBOX*));  
 }  
}

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);  
 }  
}

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));  
 }  
}

TACK comon

package tasks.common;  
  
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.targets.Target;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isClickable*;  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class ClickElement implements Task {  
  
 private final Target target;  
 private final boolean shouldScroll;  
 private final String[] values;  
  
 // Constructor que acepta un Target, un indicador de desplazamiento y un número variable de valores  
 public ClickElement(Target target, boolean shouldScroll, String... values) {  
 this.target = target;  
 this.shouldScroll = shouldScroll;  
 this.values = values;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 // Determinar el Target final basado en si se proporcionaron valores o no  
 Target finalTarget = (values == null || values.length == 0) ? target : target.of(values);  
  
 // Desplazarse al elemento si shouldScroll es verdadero  
 if (shouldScroll) {  
 actor.attemptsTo(Scroll.*to*(finalTarget));  
 }  
  
 // Esperar hasta que el elemento sea visible y clickeable  
 actor.attemptsTo(  
 WaitUntil.*the*(finalTarget, *isVisible*()).forNoMoreThan(15).seconds(),  
 WaitUntil.*the*(finalTarget, *isClickable*()).forNoMoreThan(15).seconds(),  
 // Hacer clic en el elemento  
 Click.*on*(finalTarget)  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static ClickElement on(Target target, boolean shouldScroll, String... values) {  
 return new ClickElement(target, shouldScroll, values);  
 }  
}  
  
//package tasks.common;  
//  
//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.targets.Target;  
//import net.serenitybdd.screenplay.waits.WaitUntil;  
//  
//import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.isVisible;  
//  
//public class ClickElement implements Task {  
//  
// private final Target target;  
// private final String[] values;  
//  
// // Constructor que acepta un Target y un número variable de valores  
// public ClickElement(Target target, String... values) {  
// this.target = target;  
// this.values = values;  
// }  
//  
// @Override  
// public <T extends Actor> void performAs(T actor) {  
// // Determinar el Target final basado en si se proporcionaron valores o no  
// Target finalTarget = (values == null || values.length == 0) ? target : target.of(values);  
//  
// actor.attemptsTo(  
// // Desplazarse al elemento  
// Scroll.to(finalTarget),  
// // Esperar hasta que el elemento sea visible  
// WaitUntil.the(finalTarget, isVisible()).forNoMoreThan(15).seconds(),  
// // Hacer clic en el elemento  
// Click.on(finalTarget)  
// );  
// }  
//  
// // Mé*todo estático para facilitar la creación de la tarea*  
// public static ClickElement on(Target target, String... values) {  
// return new ClickElement(target, values);  
// }  
//}

package tasks.common;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.targets.Target;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class ClickElementJs implements Task {  
 private final Target target;  
  
 public ClickElementJs(Target target) {  
 this.target = target;  
 }  
  
 public static ClickElementJs on(Target target) {  
 return *instrumented*(ClickElementJs.class, target);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 WebDriver driver = net.serenitybdd.screenplay.abilities.BrowseTheWeb.*as*(actor).getDriver();  
 WebElement element = target.resolveFor(actor);  
  
 // Usar JavaScript para hacer clic en el elemento  
 ((JavascriptExecutor) driver).executeScript("arguments[0].click();", element);  
 }  
}

package tasks.common;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.actions.Scroll;  
import net.serenitybdd.screenplay.targets.Target;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class EnterText implements Task {  
  
 private final Target target;  
 private final String value;  
  
 public EnterText(Target target, String value) {  
 this.target = target;  
 this.value = value;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 // Desplazarse al elemento  
 Scroll.*to*(target),  
 // Esperar hasta que el elemento sea visible  
 WaitUntil.*the*(target, *isVisible*()).forNoMoreThan(15).seconds(),  
 // Ingresar el valor en el elemento  
 Enter.*theValue*(value).into(target)  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static EnterText into(Target target, String value) {  
 return new EnterText(target, value);  
 }  
}

import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.targets.Target;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class ScrollTo implements Task {  
 private final Target target;  
 public ScrollTo(Target target) {  
 this.target = target;  
 }  
  
 public static ScrollTo element(Target target) {  
 return *instrumented*(ScrollTo.class, target);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 WebDriver driver = net.serenitybdd.screenplay.abilities.BrowseTheWeb.*as*(actor).getDriver();  
 WebElement element = target.resolveFor(actor);  
 // Usar JavaScript para hacer scroll al elemento  
 ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);", element);  
 }  
}

package tasks.common;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class WaitFor implements Task {  
 private final int seconds;  
  
 public WaitFor(int seconds) {  
 this.seconds = seconds;  
 }  
  
 public static WaitFor seconds(int seconds) {  
 return *instrumented*(WaitFor.class, seconds);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 try {  
 Thread.*sleep*(seconds \* 1000L); // Convertir segundos a milisegundos  
 } catch (InterruptedException e) {  
 Thread.*currentThread*().interrupt(); // Restaurar el estado de interrupción  
 throw new RuntimeException("La espera fue interrumpida", e);  
 }  
 }  
}

TASK dashboard

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.waits.WaitUntil;  
import ui.ChangeFundPage;  
import ui.DashboardPage;  
import ui.OutlookPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
  
public class ButtonModal implements Task {  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 // Esperar hasta que el botón sea visible  
 WaitUntil.*the*(DashboardPage.*MODAL\_BUTTON*, *isVisible*()).forNoMoreThan(15).seconds(),  
 // Hacer clic en el botón  
 Click.*on*(DashboardPage.*MODAL\_BUTTON*)  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static ButtonModal click() {  
 return new ButtonModal();  
 }  
}

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();  
 }  
}

package tasks.dashboard;  
  
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 MyProceduresOptions implements Task {  
  
 private final String option;  
  
 public MyProceduresOptions(String option) {  
 this.option = option;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 // Esperar hasta que el botón sea visible  
 WaitUntil.*the*(DashboardPage.*MY\_PROCEDURES\_OPTIONS\_BUTTON*.of(option), *isVisible*()).forNoMoreThan(15).seconds(),  
 // Hacer clic en el botón  
 Click.*on*(DashboardPage.*MY\_PROCEDURES\_OPTIONS\_BUTTON*.of(option))  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static MyProceduresOptions click(String option) {  
 return new MyProceduresOptions(option);  
 }  
}

TASK login

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

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*));  
 }  
}

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*));  
 }  
}

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.");  
 }  
 }  
}

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));  
 }  
 }  
}

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 PrincipalUrl implements Task {  
  
 private final String PrincipalUrl;  
 private final EnvironmentVariables environmentVariables;  
  
 public PrincipalUrl(String PrincipalUrl) {  
 this.PrincipalUrl = PrincipalUrl;  
 this.environmentVariables = ConfiguredEnvironment.*getEnvironmentVariables*();  
 }  
  
  
 public static Task CambioAFP() {  
 String PrincipalUrl = "principal.url";  
 return *instrumented*(PrincipalUrl.class, PrincipalUrl);  
 }  
  
 @Override  
 @Step("{0} Inicia la página #url")  
 public <T extends Actor> void performAs(T actor) {  
  
 String targetUrl = EnvironmentSpecificConfiguration.*from*(environmentVariables).getProperty(PrincipalUrl);  
  
 if (targetUrl == null) {  
 throw new IllegalArgumentException("URL not found in environment variables: " + PrincipalUrl);  
 } else {  
 actor.attemptsTo(Open.*url*(targetUrl));  
 }  
 }  
}

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))  
 );  
 }  
}

Task outlook

package tasks.outlook;  
  
  
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.abilities.BrowseTheWeb;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.actions.Open;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.targets.Target;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
import tasks.common.ClickElement;  
import ui.LoginOutlookPage;  
  
import java.util.ArrayList;  
  
public class LoginOutlook implements Task {  
  
 private final net.thucydides.model.util.EnvironmentVariables environmentVariables;  
 private final String email;  
 private final String password;  
  
 public LoginOutlook(String email, String password) {  
 this.environmentVariables = ConfiguredEnvironment.*getEnvironmentVariables*();  
 this.email = email;  
 this.password = password;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 WebDriver driver = BrowseTheWeb.*as*(actor).getDriver();  
  
 // Abrir una nueva pestaña  
 ((JavascriptExecutor) driver).executeScript("window.open()");  
 ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());  
 driver.switchTo().window(tabs.get(1));  
  
 // Obtener la URL de Outlook desde el archivo de configuración serenity.conf  
 String outlookUrl = EnvironmentSpecificConfiguration.*from*(environmentVariables).getProperty("environments.default.outlook.url");  
  
 if (outlookUrl == null) {  
 throw new IllegalArgumentException("URL not found in environment variables: environments.default.outlook.url");  
 }  
  
 // Navegar a Outlook  
 actor.attemptsTo(Open.*url*(outlookUrl));  
  
  
 // Esperar a que la página cargue completamente  
 actor.attemptsTo(  
 WaitUntil.*the*(LoginOutlookPage.*INP\_EMAIL\_USER*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(30).seconds()  
 );  
  
  
 // Iniciar sesión en Outlook  
 actor.attemptsTo(  
 Task.*where*("{0} inicia sesión en Outlook",  
 Enter.*theValue*(email).into(LoginOutlookPage.*INP\_EMAIL\_USER*),  
 Click.*on*(LoginOutlookPage.*NEXT\_BUTTON*),  
 WaitUntil.*the*(LoginOutlookPage.*INP\_PASSWORD\_USER*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds(),  
 Enter.*theValue*(password).into(LoginOutlookPage.*INP\_PASSWORD\_USER*),  
 Click.*on*(LoginOutlookPage.*NEXT\_BUTTON*),  
 WaitUntil.*the*(LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*.of("Sí"), WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds(),  
 Click.*on*(LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*.of("Sí"))  
 )  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static LoginOutlook login(String usuario, String clave) {  
 return new LoginOutlook(usuario, clave);  
 }  
}

package tasks.outlook;  
  
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.abilities.BrowseTheWeb;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.actions.Open;  
import net.serenitybdd.screenplay.conditions.Check;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.Keys;  
import org.openqa.selenium.WebDriver;  
import ui.LoginOutlookPage;  
import ui.OutlookPage;  
  
import java.util.ArrayList;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
import static ui.OutlookPage.\*;  
  
public class OpenOutlookAndRetrieveCode implements Task {  
  
 private final net.thucydides.model.util.EnvironmentVariables environmentVariables;  
 private final String email;  
 private final String password;  
 private final String searchText;  
  
 public OpenOutlookAndRetrieveCode(String email, String password, String searchText) {  
 this.environmentVariables = ConfiguredEnvironment.*getEnvironmentVariables*();  
 this.email = email;  
 this.password = password;  
 this.searchText = searchText;  
 }  
  
 @Override  
 @Step("{0} abre una nueva pestaña, inicia sesión en Outlook y obtiene el código de seguridad")  
 public <T extends Actor> void performAs(T actor) {  
 WebDriver driver = BrowseTheWeb.*as*(actor).getDriver();  
 String originalWindow = driver.getWindowHandle();  
  
 // Abrir una nueva pestaña  
 ((JavascriptExecutor) driver).executeScript("window.open()");  
 ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());  
 driver.switchTo().window(tabs.get(1)); // Cambiar a la nueva pestaña  
  
 // Obtener la URL de Outlook desde el archivo de configuración  
 String outlookUrl = EnvironmentSpecificConfiguration.*from*(environmentVariables).getProperty("environments.default.outlook.url");  
  
 if (outlookUrl == null) {  
 throw new IllegalArgumentException("URL not found in environment variables: environments.default.outlook.url");  
 }  
  
 // Navegar a Outlook  
 actor.attemptsTo(Open.*url*(outlookUrl));  
  
 // Iniciar sesión en Outlook  
 actor.attemptsTo(  
 Task.*where*("{0} inicia sesión en Outlook",  
 WaitUntil.*the*(LoginOutlookPage.*INP\_EMAIL\_USER*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(20).seconds(),  
 Enter.*theValue*(email).into(LoginOutlookPage.*INP\_EMAIL\_USER*),  
 Click.*on*(LoginOutlookPage.*NEXT\_BUTTON*),  
 WaitUntil.*the*(LoginOutlookPage.*INP\_PASSWORD\_USER*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds(),  
 Enter.*theValue*(password).into(LoginOutlookPage.*INP\_PASSWORD\_USER*),  
 Click.*on*(LoginOutlookPage.*NEXT\_BUTTON*),  
 WaitUntil.*the*(LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*.of("Sí"), WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds(),  
 Click.*on*(LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*.of("Sí"))  
 )  
 );  
  
 try {  
 Thread.*sleep*(10000); // Espera fija de 10 segundos  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 actor.attemptsTo(  
 WaitUntil.*the*(OutlookPage.*ACCOUNT\_1*, WebElementStateMatchers.*isPresent*())  
 .forNoMoreThan(10).seconds(),  
 Check.*whether*(OutlookPage.*ACCOUNT\_1*.resolveFor(actor).isPresent())  
 .andIfSo(  
 WaitUntil.*the*(OutlookPage.*ACCOUNT\_1*, WebElementStateMatchers.*isClickable*())  
 .forNoMoreThan(10).seconds(),  
 Click.*on*(OutlookPage.*ACCOUNT\_1*)  
 )  
 );  
  
 // Ingresar el texto de búsqueda y presionar Enter, luego esperar a que cargue la página  
 actor.attemptsTo(  
 Task.*where*("{0} ingresa el texto de búsqueda y espera a que cargue la página",  
 WaitUntil.*the*(*TXT\_SEARCH*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds(),  
 Enter.*theValue*(searchText).into(*TXT\_SEARCH*).thenHit(Keys.*ENTER*),  
 WaitUntil.*the*(*SELECT\_FIRST\_EMAIL\_1*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds()  
 )  
 );  
  
 // Esperar a que aparezcan los resultados y seleccionar el último correo  
 actor.attemptsTo(  
 Task.*where*("{0} selecciona el último correo",  
 Check.*whether*(WebElementStateMatchers.*isVisible*().matches(*SELECT\_FIRST\_EMAIL\_1*.resolveFor(actor)))  
 .andIfSo(Click.*on*(*SELECT\_FIRST\_EMAIL\_1*))  
 .otherwise(  
 WaitUntil.*the*(*SELECT\_FIRST\_EMAIL\_2*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds(),  
 Check.*whether*(WebElementStateMatchers.*isVisible*().matches(*SELECT\_FIRST\_EMAIL\_2*.resolveFor(actor)))  
 .andIfSo(Click.*on*(*SELECT\_FIRST\_EMAIL\_2*))  
 )  
 )  
 );  
  
 // Esperar a que el elemento LBL\_SECURITY\_CODE sea visible y obtener el texto  
 actor.attemptsTo(  
 WaitUntil.*the*(*LBL\_SECURITY\_CODE*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(30).seconds()  
 );  
  
 String securityCode = *LBL\_SECURITY\_CODE*.resolveFor(actor).getText();  
  
 // Regresar a la pestaña original  
 driver.close();  
 driver.switchTo().window(tabs.get(0));  
  
 // Guardar el código de seguridad en el portador  
 actor.remember("securityCode", securityCode);  
 }  
  
 public static OpenOutlookAndRetrieveCode withCredentials(String email, String password, String searchText) {  
 return *instrumented*(OpenOutlookAndRetrieveCode.class, email, password, searchText);  
 }  
}

package tasks.outlook;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.abilities.BrowseTheWeb;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.actions.Scroll;  
import net.serenitybdd.screenplay.conditions.Check;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.openqa.selenium.Keys;  
import org.openqa.selenium.WebDriver;  
import ui.LoginOutlookPage;  
import ui.OutlookPage;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Set;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
import static ui.OutlookPage.\*;  
import static ui.OutlookPage.*LBL\_SECURITY\_CODE*;  
  
public class SearchSecurityCode implements Task {  
  
 private final String asunto;  
  
 public SearchSecurityCode(String asunto) {  
 this.asunto = asunto;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 actor.attemptsTo(  
 Task.*where*("{0} inicia sesión en Outlook",  
 // Desplazarse al botón  
 Scroll.*to*(OutlookPage.*BUTTON\_SPAM*),  
 // Esperar hasta que el botón sea visible  
 WaitUntil.*the*(OutlookPage.*BUTTON\_SPAM*, *isVisible*()).forNoMoreThan(15).seconds(),  
 // Hacer clic en el botón  
 Click.*on*(OutlookPage.*BUTTON\_SPAM*)  
 )  
 );  
  
 // Ingresar el texto de búsqueda y presionar Enter, luego esperar a que cargue la página  
 actor.attemptsTo(  
 Task.*where*("{0} ingresa el texto de búsqueda y espera a que cargue la página",  
 Enter.*theValue*(asunto).into(*TXT\_SEARCH*).thenHit(Keys.*ENTER*),  
 WaitUntil.*the*(*SELECT\_FIRST\_EMAIL\_1*, *isVisible*()).forNoMoreThan(10).seconds()  
 )  
 );  
  
  
 // Esperar a que aparezcan los resultados y seleccionar el último correo  
 actor.attemptsTo(  
 Task.*where*("{0} selecciona el último correo",  
 Check.*whether*(WebElementStateMatchers.*isVisible*().matches(*SELECT\_FIRST\_EMAIL\_1*.resolveFor(actor)))  
 .andIfSo(Click.*on*(*SELECT\_FIRST\_EMAIL\_1*))  
 .otherwise(  
 WaitUntil.*the*(*SELECT\_FIRST\_EMAIL\_2*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds(),  
 Check.*whether*(WebElementStateMatchers.*isVisible*().matches(*SELECT\_FIRST\_EMAIL\_2*.resolveFor(actor)))  
 .andIfSo(Click.*on*(*SELECT\_FIRST\_EMAIL\_2*))  
 )  
 )  
 );  
  
 actor.attemptsTo(  
 WaitUntil.*the*(*WIDE\_CONTENT*, *isVisible*()).forNoMoreThan(30).seconds(),  
 Scroll.*to*(*WIDE\_CONTENT*)  
 );  
  
 actor.attemptsTo(  
  
 // Esperar hasta que el elemento sea visible  
 WaitUntil.*the*(*LBL\_SECURITY\_CODE*, *isVisible*()).forNoMoreThan(30).seconds(),  
 // Desplazarse al elemento LBL\_SECURITY\_CODE  
 Scroll.*to*(*LBL\_SECURITY\_CODE*)  
 );  
  
 //Ontener el codigo  
 String securityCode = *LBL\_SECURITY\_CODE*.resolveFor(actor).getText();  
  
 // Guardar el código de seguridad  
 actor.remember("securityCode", securityCode);  
  
 //Obtener el driver  
 WebDriver driver = BrowseTheWeb.*as*(actor).getDriver();  
  
 // Obtener todos los identificadores de ventanas abiertas  
 Set<String> windowHandles = driver.getWindowHandles();  
 List<String> windowHandlesList = new ArrayList<>(windowHandles);  
  
 // Identificar la primera pestaña  
 String firstWindowHandle = windowHandlesList.get(0);  
  
 // Cerrar la segunda pestaña  
 driver.close();  
  
 // Cambiar el control de vuelta a la primera pestaña  
 driver.switchTo().window(firstWindowHandle);  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static SearchSecurityCode With(String asunto) {  
 return new SearchSecurityCode(asunto);  
 }  
}

TASK restPassword

package tasks.resetPassword;  
  
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.ResetPasswordPage;  
  
public class ButtonContinue implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonContinue.class);  
  
 @Override  
 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 medio segundo antes de inciar  
 if (ResetPasswordPage.*CONTINUE\_BUTTON*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ResetPasswordPage.*CONTINUE\_BUTTON*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ResetPasswordPage.*CONTINUE\_BUTTON*)  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El botón Continuar no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el botón Continuar. 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 botón Continuar después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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.LoginOutlookPage;  
import ui.ResetPasswordPage;  
  
public class ButtonNext implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonNext.class);  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 15000; // 10 segundos  
 boolean clicked = false;  
  
 while (System.*currentTimeMillis*() < endTime && !clicked) {  
 try {  
 Thread.*sleep*(1000); // Espera medio segundo antes de reintentar  
 if (LoginOutlookPage.*NEXT\_BUTTON*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(LoginOutlookPage.*NEXT\_BUTTON*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(LoginOutlookPage.*NEXT\_BUTTON*)  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El botón Next no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el botón Next. 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 botón Next después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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.ResetPasswordPage;  
  
public class ButtonOk implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonOk.class);  
  
 public static ButtonOk click() {  
 return new ButtonOk();  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 10000; // 10 segundos  
 boolean clicked = false;  
  
 while (System.*currentTimeMillis*() < endTime && !clicked) {  
 try {  
 if (ResetPasswordPage.*OK\_BUTTON*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ResetPasswordPage.*OK\_BUTTON*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ResetPasswordPage.*OK\_BUTTON*)  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El botón OK no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el botón OK. 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 botón OK después de 10 segundos.");  
 }  
 }  
}

ackage tasks.resetPassword;  
  
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.ResetPasswordPage;  
  
public class ButtonResetPassword implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonResetPassword.class);  
  
 public static ButtonResetPassword click() {  
 return new ButtonResetPassword();  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 10000; // 10 segundos  
 boolean clicked = false;  
  
 while (System.*currentTimeMillis*() < endTime && !clicked) {  
 try {  
 if (ResetPasswordPage.*BTN\_RECOVER\_PASSWORD*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ResetPasswordPage.*BTN\_RECOVER\_PASSWORD*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ResetPasswordPage.*BTN\_RECOVER\_PASSWORD*)  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El botón Recover Password no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el botón Recover Password. 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 botón Recover Password después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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.ResetPasswordPage;  
  
public class ButtonSearch implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonSearch.class);  
  
 @Override  
 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 medio segundo antes de iniciar  
 if (ResetPasswordPage.*BTN\_SEARCH*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ResetPasswordPage.*BTN\_SEARCH*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ResetPasswordPage.*BTN\_SEARCH*)  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El botón Search no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el botón Search. 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 botón Search después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.LoginOutlookPage;  
import ui.ResetPasswordPage;  
  
public class ButtonSendEmail implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonSendEmail.class);  
 private final String clickSendEMail;  
  
 public ButtonSendEmail(String clickSendEMail) {  
 this.clickSendEMail = clickSendEMail;  
 }  
  
 public static ButtonSendEmail withData(String clickSendEMail) {  
 return new ButtonSendEmail(clickSendEMail);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 15000; // 10 segundos  
 boolean clicked = false;  
  
 while (System.*currentTimeMillis*() < endTime && !clicked) {  
 try {  
 Thread.*sleep*(1000); // Espera medio segundo antes de reintentar  
 if (LoginOutlookPage.*BTN\_SEND\_EMAIL*.of(clickSendEMail).resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(LoginOutlookPage.*BTN\_SEND\_EMAIL*.of(clickSendEMail), WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Scroll.*to*(LoginOutlookPage.*BTN\_SEND\_EMAIL*.of(clickSendEMail)),  
 Click.*on*(LoginOutlookPage.*BTN\_SEND\_EMAIL*.of(clickSendEMail))  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El botón Send Email no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el botón Send Email. 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 botón Send Email después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
import net.serenitybdd.annotations.Step;  
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.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.ResetPasswordPage;  
  
public class ButtonShowContent implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonShowContent.class);  
  
 public static ButtonShowContent now() {  
 return Tasks.*instrumented*(ButtonShowContent.class);  
 }  
  
 @Override  
 @Step("Hacer clic en el botón Show Content si existe")  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 15000; // 10 segundos  
 boolean clicked = false;  
  
 while (System.*currentTimeMillis*() < endTime && !clicked) {  
 try {  
 Thread.*sleep*(500); // Espera inicial de 2000 milisegundos  
 if (ResetPasswordPage.*BTN\_SHOW\_CONTENT*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ResetPasswordPage.*BTN\_SHOW\_CONTENT*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ResetPasswordPage.*BTN\_SHOW\_CONTENT*)  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("El botón Show Content no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el botón Show Content. 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 botón Show Content después de 15 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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.ResetPasswordPage;  
  
public class ButtonUnderStood implements Task {  
  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ResetPasswordPage.*BTN\_UNDERSTOOD*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds(),  
 WaitUntil.*the*(ResetPasswordPage.*BTN\_UNDERSTOOD*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(ResetPasswordPage.*BTN\_UNDERSTOOD*));  
 }  
}

package tasks.resetPassword;  
  
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.Click;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.LoginOutlookPage;  
  
public class ButtonYesNotSession implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ButtonYesNotSession.class);  
 private final String option;  
  
 public ButtonYesNotSession(String option) {  
 this.option = option;  
 }  
  
 public static Performable withData(String option) {  
 return Tasks.*instrumented*(ButtonYesNotSession.class, option);  
 }  
  
 @Override  
 @Step("Selecciona la opción {0} recordar sesión")  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 10000; // 10 segundos  
 boolean clicked = false;  
  
 while (System.*currentTimeMillis*() < endTime && !clicked) {  
 try {  
 if (LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*.of(option).resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*.of(option), WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Click.*on*(LoginOutlookPage.*BTN\_YES\_NOT\_OPTION*.of(option))  
 );  
 clicked = true;  
 } else {  
 *LOGGER*.info("La opción {} no está visible inicialmente.", option);  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en la opción {}. Reintentando...", option, 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 la opción {} después de 10 segundos.", option);  
 }  
 }  
}

package tasks.resetPassword;  
  
import net.serenitybdd.annotations.Step;  
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.Scroll;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.ResetPasswordPage;  
  
public class ClickLinkChoose implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(ClickLinkChoose.class);  
  
 public static ClickLinkChoose now() {  
 return Tasks.*instrumented*(ClickLinkChoose.class);  
 }  
  
 @Override  
 @Step("Hacer clic en el enlace IMG\_LINK\_URL si existe")  
 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 (ResetPasswordPage.*IMG\_LINK\_URL*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(WaitUntil.*the*(ResetPasswordPage.*IMG\_LINK\_URL*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(), Scroll.*to*(ResetPasswordPage.*IMG\_LINK\_URL*));  
 actor.attemptsTo(WaitUntil.*the*(ResetPasswordPage.*IMG\_LINK\_URL*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(), Click.*on*(ResetPasswordPage.*IMG\_LINK\_URL*));  
 clicked = true;  
 } else {  
 *LOGGER*.info("El enlace IMG\_LINK\_URL no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de hacer clic en el enlace IMG\_LINK\_URL. 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 enlace IMG\_LINK\_URL después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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 org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.ResetPasswordPage;  
  
public class InputNewPassword implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(InputNewPassword.class);  
 private final String enterNewPassword;  
  
 public InputNewPassword(String enterNewPassword) {  
 this.enterNewPassword = enterNewPassword;  
 }  
  
 public static InputNewPassword withData(String enterNewPassword) {  
 return new InputNewPassword(enterNewPassword);  
 }  
  
 @Override  
 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 (ResetPasswordPage.*INP\_NEW\_PASSWORD*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(ResetPasswordPage.*INP\_NEW\_PASSWORD*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Enter.*theValue*(enterNewPassword).into(ResetPasswordPage.*INP\_NEW\_PASSWORD*)  
 );  
 entered = true;  
 } else {  
 *LOGGER*.info("El campo de entrada de nueva contraseña no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de ingresar la nueva contraseña. 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 la nueva contraseña después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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 org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.LoginOutlookPage;  
  
public class InputPasswordEmail implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(InputPasswordEmail.class);  
 private final String enterPasswordEmail;  
  
 public InputPasswordEmail(String enterPasswordEmail) {  
 this.enterPasswordEmail = enterPasswordEmail;  
 }  
  
 public static InputPasswordEmail withData(String enterPasswordEmail) {  
 return new InputPasswordEmail(enterPasswordEmail);  
 }  
  
 @Override  
 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 (LoginOutlookPage.*INP\_PASSWORD\_USER*.resolveFor(actor).isVisible()) {  
 actor.attemptsTo(  
 WaitUntil.*the*(LoginOutlookPage.*INP\_PASSWORD\_USER*, WebElementStateMatchers.*isClickable*()).forNoMoreThan(15).seconds(),  
 Enter.*theValue*(enterPasswordEmail).into(LoginOutlookPage.*INP\_PASSWORD\_USER*)  
 );  
 entered = true;  
 } else {  
 *LOGGER*.info("El campo de entrada de contraseña no está visible inicialmente.");  
 }  
 } catch (Exception e) {  
 *LOGGER*.warn("Intento fallido de ingresar la contraseña. 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 la contraseña después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.openqa.selenium.Keys;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.OutlookPage;  
import ui.ResetPasswordPage;  
  
import static ui.OutlookPage.*SELECT\_FIRST\_EMAIL\_1*;  
import static ui.OutlookPage.*TXT\_SEARCH*;  
  
public class InputSearchEmail implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(InputSearchEmail.class);  
 private final String enterSearchEmail;  
  
 public InputSearchEmail(String enterSearchEmail) {  
 this.enterSearchEmail = enterSearchEmail;  
 }  
  
 public static InputSearchEmail withData(String enterSearchEmail) {  
 return new InputSearchEmail(enterSearchEmail);  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 long endTime = System.*currentTimeMillis*() + 10000; // 10 segundos  
 boolean entered = false;  
  
  
 try {  
 Thread.*sleep*(10000); // Espera fija de 10 segundos  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
 actor.attemptsTo(  
 Click.*on*(OutlookPage.*ACCOUNT\_1*)  
 );  
  
 // Ingresar el texto de búsqueda y presionar Enter, luego esperar a que cargue la página  
 actor.attemptsTo(  
 Task.*where*("{0} ingresa el texto de búsqueda y espera a que cargue la página",  
 WaitUntil.*the*(*TXT\_SEARCH*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds(),  
 Click.*on*(*TXT\_SEARCH*),  
 Enter.*theValue*(enterSearchEmail).into(*TXT\_SEARCH*).thenHit(Keys.*ENTER*),  
 WaitUntil.*the*(*SELECT\_FIRST\_EMAIL\_1*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(15).seconds()  
 )  
 );  
  
// while (System.currentTimeMillis() < endTime && !entered) {  
// try {  
//  
// if (ResetPasswordPage.INP\_SEARCH\_TITLE.resolveFor(actor).isVisible()) {  
// actor.attemptsTo(  
// WaitUntil.the(ResetPasswordPage.INP\_SEARCH\_TITLE, WebElementStateMatchers.isClickable()).forNoMoreThan(15).seconds(),  
// Enter.theValue(enterSearchEmail).into(ResetPasswordPage.INP\_SEARCH\_TITLE)  
// );  
// entered = true;  
// } else {  
// LOGGER.info("El campo de entrada de búsqueda de correo no está visible inicialmente.");  
// }  
// } catch (Exception e) {  
// LOGGER.warn("Intento fallido de ingresar el correo de búsqueda. 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 correo de búsqueda después de 10 segundos.");  
 }  
 }  
}

package tasks.resetPassword;  
  
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.Click;  
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.LoginOutlookPage;  
  
public class InputUserEmail implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(InputUserEmail.class);  
 private final String enterUserEmail;  
  
 public InputUserEmail(String enterUserEmail) {  
 this.enterUserEmail = enterUserEmail;  
 }  
  
 public static Performable withData(String enterUserEmail) {  
 return Tasks.*instrumented*(InputUserEmail.class, enterUserEmail);  
 }  
  
 @Override  
 @Step("{0} ingresa usuario email")  
 public <T extends Actor> void performAs(T actor) {  
  
 // Iniciar sesión en Outlook  
 actor.attemptsTo(  
 Task.*where*("{0} inicia sesión en Outlook",  
 WaitUntil.*the*(LoginOutlookPage.*INP\_EMAIL\_USER*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(20).seconds(),  
 Enter.*theValue*(enterUserEmail).into(LoginOutlookPage.*INP\_EMAIL\_USER*),  
 Click.*on*(LoginOutlookPage.*NEXT\_BUTTON*)  
// WaitUntil.the(LoginOutlookPage.INP\_PASSWORD\_USER, WebElementStateMatchers.isVisible()).forNoMoreThan(10).seconds(),  
// Enter.theValue(password).into(LoginOutlookPage.INP\_PASSWORD\_USER),  
// Click.on(LoginOutlookPage.NEXT\_BUTTON),  
// WaitUntil.the(LoginOutlookPage.BTN\_YES\_NOT\_OPTION.of("Sí"), WebElementStateMatchers.isVisible()).forNoMoreThan(10).seconds(),  
// Click.on(LoginOutlookPage.BTN\_YES\_NOT\_OPTION.of("Sí"))  
 )  
 );  
  
  
 long endTime = System.*currentTimeMillis*() + 10000; // 10 segundos  
 boolean entered = false;  
  
  
 try {  
 Thread.*sleep*(5000); // Espera fija de 10 segundos  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
  
//  
// while (System.currentTimeMillis() < endTime && !entered) {  
// try {  
// if (LoginOutlookPage.INP\_EMAIL\_USER.resolveFor(actor).isVisible()) {  
// actor.attemptsTo(  
// WaitUntil.the(LoginOutlookPage.INP\_EMAIL\_USER, WebElementStateMatchers.isClickable()).forNoMoreThan(15).seconds(),  
// Enter.theValue(enterUserEmail).into(LoginOutlookPage.INP\_EMAIL\_USER)  
// );  
// entered = true;  
// } else {  
// LOGGER.info("El campo de entrada de correo de usuario no está visible inicialmente.");  
// }  
// } catch (Exception e) {  
// LOGGER.warn("Intento fallido de ingresar el correo de 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 correo de usuario después de 10 segundos.");  
// }  
 }  
}

package tasks.resetPassword;  
  
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.abilities.BrowseTheWeb;  
import net.serenitybdd.screenplay.actions.Open;  
import net.serenitybdd.screenplay.actions.Switch;  
import net.thucydides.model.util.EnvironmentVariables;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
  
import java.util.ArrayList;  
  
import static net.serenitybdd.screenplay.Tasks.*instrumented*;  
  
public class OpenUrlOutlook implements Task {  
  
 private final String url;  
 private final EnvironmentVariables environmentVariables;  
  
 public OpenUrlOutlook(String url) {  
 this.url = url;  
 this.environmentVariables = ConfiguredEnvironment.*getEnvironmentVariables*();  
 }  
  
 public static Task loginOutlook() {  
 String url = "outlook.url";  
 return *instrumented*(OpenUrlOutlook.class, url);  
 }  
  
 @Override  
 @Step("{0} Inicia la página #url")  
 public <T extends Actor> void performAs(T actor) {  
  
 WebDriver driver = BrowseTheWeb.*as*(actor).getDriver();  
 String originalWindow = driver.getWindowHandle();  
  
 // Abrir una nueva pestaña  
 ((JavascriptExecutor) driver).executeScript("window.open()");  
 ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());  
 driver.switchTo().window(tabs.get(1)); // Cambiar a la nueva pestaña  
  
 // Obtener la URL de Outlook desde el archivo de configuración  
 String outlookUrl = EnvironmentSpecificConfiguration.*from*(environmentVariables).getProperty("environments.default.outlook.url");  
  
 if (outlookUrl == null) {  
 throw new IllegalArgumentException("URL not found in environment variables: environments.default.outlook.url");  
 }  
  
 // Navegar a Outlook  
 actor.attemptsTo(Open.*url*(outlookUrl));  
  
  
// String targetUrl = EnvironmentSpecificConfiguration.from(environmentVariables).getProperty(url);  
//  
// if (targetUrl == null) {  
// throw new IllegalArgumentException("URL not found in environment variables: " + url);  
// } else {  
// actor.attemptsTo(  
// Switch.toNewWindow(),  
// Open.url(targetUrl));  
// }  
 }  
}

package tasks.resetPassword;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.conditions.Check;  
import net.serenitybdd.screenplay.matchers.WebElementStateMatchers;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import ui.ResetPasswordPage;  
  
import static ui.OutlookPage.*SELECT\_FIRST\_EMAIL\_1*;  
import static ui.OutlookPage.*SELECT\_FIRST\_EMAIL\_2*;  
  
public class SelectFirstEmail implements Task {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SelectFirstEmail.class);  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
  
 // Esperar a que aparezcan los resultados y seleccionar el último correo  
 actor.attemptsTo(  
 Task.*where*("selecciona el último correo",  
 Check.*whether*(WebElementStateMatchers.*isVisible*().matches(*SELECT\_FIRST\_EMAIL\_1*.resolveFor(actor)))  
 .andIfSo(Click.*on*(*SELECT\_FIRST\_EMAIL\_1*))  
 .otherwise(  
 WaitUntil.*the*(*SELECT\_FIRST\_EMAIL\_2*, WebElementStateMatchers.*isVisible*()).forNoMoreThan(10).seconds(),  
 Check.*whether*(WebElementStateMatchers.*isVisible*().matches(*SELECT\_FIRST\_EMAIL\_2*.resolveFor(actor)))  
 .andIfSo(Click.*on*(*SELECT\_FIRST\_EMAIL\_2*))  
 )  
 )  
 );  
  
// long endTime = System.currentTimeMillis() + 15000; // 15 segundos  
// boolean clicked = false;  
//  
// while (System.currentTimeMillis() < endTime && !clicked) {  
// try {  
// Thread.sleep(500); // Espera inicial de 500 milisegundos  
// if (ResetPasswordPage.SELECT\_FIRST\_EMAIL.of(String.valueOf(2)).resolveFor(actor).isVisible()) {  
// actor.attemptsTo(WaitUntil.the(ResetPasswordPage.SELECT\_FIRST\_EMAIL.of(String.valueOf(2)), WebElementStateMatchers.isClickable()).forNoMoreThan(15).seconds(), Click.on(ResetPasswordPage.SELECT\_FIRST\_EMAIL.of(String.valueOf(2))));  
// clicked = true;  
// } else {  
// actor.attemptsTo(WaitUntil.the(ResetPasswordPage.SELECT\_FIRST\_EMAIL.of(String.valueOf(1)), WebElementStateMatchers.isClickable()).forNoMoreThan(15).seconds(), Click.on(ResetPasswordPage.SELECT\_FIRST\_EMAIL.of(String.valueOf(1))));  
// clicked = true;  
// }  
// } catch (Exception e) {  
// LOGGER.warn("Intento fallido de hacer clic en el primer correo electrónico. 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 primer correo electrónico después de 15 segundos.");  
// }  
 }  
}

package tasks.resetPassword;  
  
import net.serenitybdd.annotations.Step;  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.Tasks;  
import net.serenitybdd.screenplay.abilities.BrowseTheWeb;  
import net.serenitybdd.screenplay.actions.Switch;  
import org.openqa.selenium.WebDriver;  
  
import java.util.ArrayList;  
  
public class SwitchToTab implements Task {  
 private final int tabIndex;  
  
 public SwitchToTab(int tabIndex) {  
 this.tabIndex = tabIndex;  
 }  
  
 public static SwitchToTab to(int tabIndex) {  
 return Tasks.*instrumented*(SwitchToTab.class, tabIndex);  
 }  
  
 @Override  
 @Step("Cambiar a la pestaña número {0}")  
 public <T extends Actor> void performAs(T actor) {  
 WebDriver driver = BrowseTheWeb.*as*(actor).getDriver();  
 ArrayList<String> tabs = new ArrayList<>(driver.getWindowHandles());  
 if (tabs.size() > tabIndex) {  
 actor.attemptsTo(Switch.*toWindow*(tabs.get(tabIndex)));  
 } else {  
 throw new IllegalStateException("No hay suficientes pestañas abiertas para cambiar a la pestaña número " + tabIndex);  
 }  
 }  
}

TASK voluntaryContributions

package tasks.voluntaryContributions;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import tasks.common.ClickElement;  
import tasks.common.EnterText;  
import ui.VoluntaryContributionsPage;  
  
public class SelectOriginOfMoney implements Task {  
  
 private final String valorOrigen;  
  
 public SelectOriginOfMoney(String valorOrigen) {  
 this.valorOrigen = valorOrigen;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 String[] origenes = valorOrigen.split(",");  
  
 for (int i = 0; i < origenes.length; i++) {  
 String[] partes = origenes[i].split("-");  
 String tipoOrigen = partes[0].trim();  
 String banco = partes[1].trim();  
  
 // Seleccionar el tipo de origen  
 actor.attemptsTo(  
 ClickElement.*on*(VoluntaryContributionsPage.*ORIGIN\_SELECT*, true, String.*valueOf*(i + 1)),  
 ClickElement.*on*(VoluntaryContributionsPage.*ORIGINOPTIONS\_ITEM*, true, tipoOrigen, String.*valueOf*(i + 1))  
 );  
  
 // Ingresar el banco  
 actor.attemptsTo(  
 EnterText.*into*(VoluntaryContributionsPage.*BANK\_INPUT*.of(String.*valueOf*(i + 1)), banco)  
 );  
  
 // Hacer clic en "Agregar otro" si es necesario  
 if (i < origenes.length - 1) {  
 actor.attemptsTo(  
 ClickElement.*on*(VoluntaryContributionsPage.*ADD\_OTHER\_BUTTON*, true)  
 );  
 }  
 }  
 }  
  
 public static SelectOriginOfMoney withValues(String valorOrigen) {  
 return new SelectOriginOfMoney(valorOrigen);  
 }  
}

TASK withdrawVoluntaryContributions

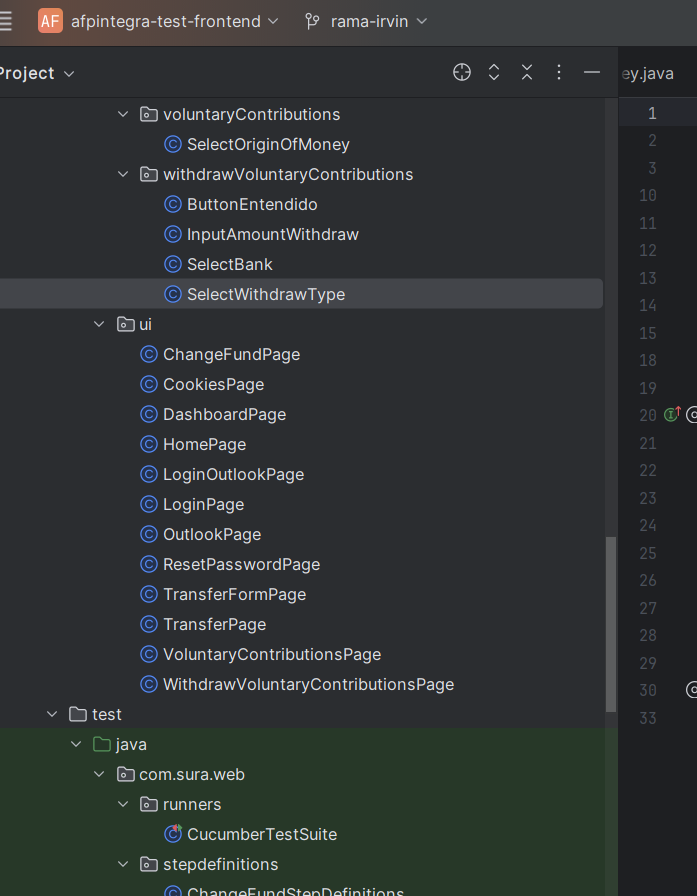
package tasks.withdrawVoluntaryContributions;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.WithdrawVoluntaryContributionsPage;  
  
import static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

package tasks.withdrawVoluntaryContributions;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Enter;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.WithdrawVoluntaryContributionsPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class InputAmountWithdraw implements Task {  
  
 private final String amount;  
  
 public InputAmountWithdraw(String amount) {  
 this.amount = amount;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 // Esperar hasta que el campo de entrada sea visible  
 WaitUntil.*the*(WithdrawVoluntaryContributionsPage.*AMOUNT\_WITHDRAWTYPE*, *isVisible*()).forNoMoreThan(15).seconds(),  
 // Ingresar el valor en el campo de entrada  
 Enter.*theValue*(amount).into(WithdrawVoluntaryContributionsPage.*AMOUNT\_WITHDRAWTYPE*)  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static InputAmountWithdraw withAmount(String amount) {  
 return new InputAmountWithdraw(amount);  
 }  
}

package tasks.withdrawVoluntaryContributions;  
  
public class SelectBank {  
}

package tasks.withdrawVoluntaryContributions;  
  
import net.serenitybdd.screenplay.Actor;  
import net.serenitybdd.screenplay.Task;  
import net.serenitybdd.screenplay.actions.Click;  
import net.serenitybdd.screenplay.waits.WaitUntil;  
import ui.WithdrawVoluntaryContributionsPage;  
  
import static net.serenitybdd.screenplay.matchers.WebElementStateMatchers.*isVisible*;  
  
public class SelectWithdrawType implements Task {  
  
 private final String option;  
  
 public SelectWithdrawType(String option) {  
 this.option = option;  
 }  
  
 @Override  
 public <T extends Actor> void performAs(T actor) {  
 actor.attemptsTo(  
 // Esperar hasta que el botón sea visible  
 WaitUntil.*the*(WithdrawVoluntaryContributionsPage.*CHKBUTTON\_WITHDRAWTYPE*.of(option), *isVisible*()).forNoMoreThan(15).seconds(),  
 // Hacer clic en el botón  
 Click.*on*(WithdrawVoluntaryContributionsPage.*CHKBUTTON\_WITHDRAWTYPE*.of(option))  
 );  
 }  
  
 // Mé*todo estático para facilitar la creación de la tarea*  
public static SelectWithdrawType click(String option) {  
 return new SelectWithdrawType(option);  
 }  
}

USER INTERFACE



package ui;  
  
import net.serenitybdd.core.pages.ResolvableElement;  
import net.serenitybdd.screenplay.targets.Target;  
  
public class ChangeFundPage {  
  
 public static final Target *YES\_LETS\_START\_BUTTON* = Target.*the*("Sí, empecemos")  
 .locatedBy("//div[text()='Sí, empecemos']/..");  
  
 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}']");  
  
 public static final Target *ACCEPT\_TERMS\_CHECKBOX* = Target.*the*("Aceptar términos y condiciones")  
 .locatedBy("//input[@id='checkAcceptTerms']/..");  
  
 public static final Target *CONTINUE\_BUTTON\_1* = Target.*the*("Continuar")  
 .locatedBy("(//button//div[text()='Continuar'])[1]/..");  
  
 public static final Target *CONTINUE\_BUTTON\_2* = Target.*the*("Continuar")  
 .locatedBy("(//button//div[text()='Continuar'])[2]/..");  
  
 public static final Target *SECURITY\_CODE\_FIELD* = Target.*the*("Código de seguridad nro{0}")  
 .locatedBy("//input[@id='inputtokencode{0}']");  
  
 public static final Target *VALIDATE\_CODE\_BUTTON* = Target.*the*("Válidar código")  
 .locatedBy("//button//div[text()='Validar código']/..");  
  
 public static final Target *H5\_LABEL* = Target.*the*("Válidar el téxto {0}")  
 .locatedBy("//h5[text()='{0}']");  
  
 public static final Target *LBL\_MESSAGE\_CHANGE\_FUND* = Target.*the*("Válidar el mensaje {0}")  
 .locatedBy("(//p[contains(text(),'{0}')])[1]");  
  
 public static final Target *BUTTON\_CAMBIAR\_FONDO* = Target.*the*("Botón cambiar fondo")  
 .locatedBy("(//div[text()='Cambiar tu Tipo de Fondo'])[1]");  
 //(//div[@class='CardComponent\_c\_card\_\_icon\_container\_\_yPujY'])[24]//\*[@class='FormalitiesHomeSection\_s\_formalities\_home\_\_modal\_icon\_\_6GyAH']  
//(//\*[local-name()='svg' and contains(@class, 'FormalitiesHomeSection')])[24]  
 public static final Target *BUTTON\_INICIAR\_TRAMITE* = Target.*the*("Botón iniciar trámite")  
 .locatedBy("//div[contains(text(),'Iniciar trámite')]/parent::a");  
  
  
}

package ui;  
  
import net.serenitybdd.core.annotations.findby.By;  
import net.serenitybdd.screenplay.targets.Target;  
  
public class CookiesPage {  
 public static final Target *COOKIES\_ACCEPT* = Target.*the*("Accept Cookies")  
 .located(By.*id*("btn\_accept\_all\_cookies"));  
 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}']");  
 public static final Target *PASSWORD\_FIELD* = Target.*the*("Enter Password")  
 .located(org.openqa.selenium.By.*id*("inpPassword"));  
 public static final Target *USERNAME\_FIELD* = Target.*the*("Enter Username")  
 .located(org.openqa.selenium.By.*id* ("inpDocumentNumber"));  
 public static final Target *LOGIN\_BUTTON* = Target.*the*("Click on the login button")  
 .located(By.*xpath*("//button[@type='submit']"));  
  
 public static final Target *MESSAGE\_WELCOME\_LOGIN* = Target.*the*("Message Welcome")  
 .locatedBy("//div[@class='i-subtitle']");  
  
 public static final Target *FIELD\_INVALID\_MESSAGE\_USER* = Target.*the*("Message Invalid Field User")  
 .located(By.*xpath*("//\*[@class='Input\_c\_input\_\_message\_\_TdP1e']/span"));  
 public static final Target *FIELD\_INVALID\_MESSAGE\_MODAL\_USER* = Target.*the*("Message Invalid Modal User")  
 .located(By.*xpath*("//\*[@class='i-spacing-4x i-body']"));  
 public static final Target *FIELD\_INVALID\_MESSAGE\_PASSWORD* = Target.*the*("Message Invalid Field Password")  
 .located(By.*xpath*("//\*[@class='i-text-error i-caption']"));  
 public static final Target *ALERT\_INVALID\_MESSAGE\_PASSWORD* = Target.*the*("Message Invalid Alert Password")  
 .located(By.*xpath*("//\*[@class='Alert\_c\_alert\_\_message\_\_bzoiK i-small i-text-left i-text-gray']"));  
  
}

package ui;  
  
import net.serenitybdd.screenplay.targets.Target;  
  
public class DashboardPage {  
  
 public static final Target *MY\_PROCEDURES\_BUTTON* = Target.*the*("Mis trámites")  
 .locatedBy("//a[@href='/mis-tramites']");  
  
 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)]");  
  
 public static final Target *MODAL\_BUTTON* = Target.*the*("Botón del modal")  
 .locatedBy("//div[contains(@class,'IsOpen')]//button");  
  
 public static final Target *MY\_PROCEDURES\_OPTIONS\_BUTTON* = Target.*the*("Mis trámites opción: {0}")  
 .locatedBy("//div[text()='{0}']/../..//button[not(@disabled)]");  
}

package ui;  
  
import net.serenitybdd.screenplay.targets.Target;  
  
public class HomePage {  
  
 public static final Target *BUTTON\_HAZTE\_CLIENTE* = Target.*the*("clic en botón hazte cliente")  
 .locatedBy("//button[@id='home\_\_show\_modal\_\_become\_a\_customer']");  
  
 public static final Target *BUTTON\_CAMBIATE\_YA* = Target.*the*("clic en botón cámbiate ya")  
 .locatedBy("//a[@id='home\_\_modal\_\_change\_to\_integra']");  
  
  
}

package ui;  
  
import net.serenitybdd.core.annotations.findby.By;  
import net.serenitybdd.screenplay.targets.Target;  
  
public class LoginOutlookPage {  
  
 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 *BTN\_SEND\_EMAIL* = Target.*the*("send email button")  
 .locatedBy("//\*[contains(text(), '{0}')]/..");  
 public static final Target *INP\_PASSWORD\_USER* = Target.*the*("Enter Password Email Outlook")  
 .locatedBy("//input[@name='passwd']");  
 public static final Target *BTN\_YES\_NOT\_OPTION* = Target.*the*("Click en el botón '{0}' recordar sesión")  
 .locatedBy("//input[@value='{0}']");  
  
}

package ui;  
  
import net.serenitybdd.core.annotations.findby.By;  
import net.serenitybdd.screenplay.targets.Target;  
  
public class LoginPage {  
 // Cambia el selector a By.cssSelector  
 public static final Target *DOCUMENT\_TYPE\_DROPDOWN* = Target.*the*("dropdown de tipo de documento")  
 .located(By.*cssSelector*(".Dropdown\_c\_dropdown\_\_relative\_\_UtAAQ"));  
  
 // Asegúrate de que este selector también es correcto  
 public static final Target *CLICK\_OPTION\_TYPE\_DROPDOWN* = Target.*the*("opción del dropdown")  
 .locatedBy("//li[@data-keyprop='{0}']");  
  
 public static final Target *PASSWORD\_FIELD* = Target.*the*("Aceptar Cookies")  
 .located(org.openqa.selenium.By.*id*("inpDocumentNumber"));  
 public static final Target *USERNAME\_FIELD* = Target.*the*("Aceptar Cookies")  
 .located(org.openqa.selenium.By.*id* ("inpPassword"));  
  
 public static final Target *LOGIN\_BUTTON* = Target.*the*("click dropdown de tipo de documento")  
 .located(By.*className*("Button\_c\_button\_\_text\_\_1E4uC"));  
  
 public static final Target *MENSAJE\_PASSWORD\_ERROR* = Target.*the*("Message Invalid Password")  
 .located(By.*xpath*("//p[contains(@class,'Alert\_c\_alert')]"));  
  
 public static final Target *VALIDAR\_MENSAJE\_DATOSINCORRECT* = Target.*the*("Mensaje por usuario incorrecto")  
 .locatedBy("//li[contains(text(),'Puede que no hayas ingresado correctamente')]");  
  
}

package ui;  
  
import net.serenitybdd.screenplay.targets.Target;  
  
public class OutlookPage {  
  
 public static final Target *LBL\_SECURITY\_CODE* = Target.*the*("security code")  
 .locatedBy("//td[@class='x\_code']//div//p");  
  
 public static final Target *TXT\_SEARCH* = Target.*the*("búsqueda")  
 .locatedBy("//input[@id='topSearchInput']");  
  
 public static final Target *SELECT\_FIRST\_EMAIL\_1* = Target.*the*("Seleciona el correo mas reciente")  
 .locatedBy("//div[@title='Todos los resultados']/ancestor::div[@data-animatable='true']//following-sibling::div[@id]");  
  
 public static final Target *SELECT\_FIRST\_EMAIL\_2* = Target.*the*("Seleciona el correo mas reciente")  
 .locatedBy("//div[@title='Resultados principales']/ancestor::div[@data-animatable='true']//following-sibling::div[@id]");  
  
 public static final Target *BUTTON\_SPAM* = Target.*the*("SPAM")  
 .locatedBy("//span[text()='Correo no deseado']/parent::div");  
  
 public static final Target *WIDE\_CONTENT* = Target.*the*("Contenido")  
 .locatedBy("//div[@class='wide-content-host']");  
  
 public static final Target *ACCOUNT\_1* = Target.*the*("Primera cuenta")  
 .locatedBy("(//div[@role='listitem']//img)[1]");  
  
  
}

package ui;  
  
import net.serenitybdd.core.annotations.findby.By;  
import net.serenitybdd.screenplay.targets.Target;  
  
public class ResetPasswordPage {  
  
 public static final Target *BTN\_RECOVER\_PASSWORD* = Target.*the*("Hacer el click en el Botón ¿Olvidé mi Contraseña?")  
 .located(By.*xpath*("(//\*[@class='i-text-aqua i-cursor-pointer i-small-medium'])[1]"));  
 public static final Target *MESSAGE\_EMAIL\_VERIFICATION* = Target.*the*("Mensaje de Verificación del Correo de Electrónico")  
 .located(By.*xpath*("//div[@class='ContainerWithPersonsComponents\_c\_container\_with\_persons\_\_gGqAd i-text-center i-text-white']//div[@class='i-spacing-4x i-body i-text-white']"));  
 public static final Target *BTN\_SEARCH* = Target.*the*("Hacer click en el boton buscar")  
 .located(By.*xpath*("//\*[@title = 'Buscar']"));  
 public static final Target *SELECT\_FIRST\_EMAIL* = Target.*the*("Seleciona el correo mas reciente")  
 .locatedBy("(//div[@title='Todos los resultados' or @title='Resultados principales']/ancestor::div[@data-animatable='true']//following-sibling::div[@id])[{0}]");  
 public static final Target *INP\_NEW\_PASSWORD* = Target.*the*("Ingresa la nueva Contraseña")  
 .located(By.*id*("inpNewPassword"));  
 public static final Target *MESSAGE\_HAPPY\_VERIFICATION\_EMAIL* = Target.*the*("Mensaje de verificación del cambio de contraseña exitoso")  
 .located(By.*xpath*("//span[@class = 'i-body-medium i-text-blue']"));  
 public static final Target *BTN\_UNDERSTOOD* = Target.*the*("Mensaje de verificación del cambio de contraseña exitoso")  
 .located(By.*xpath*("//\*[@class = 'Button\_c\_button\_\_I\_x2\_ Button\_c\_button\_\_SizeMedium\_\_lNORL Button\_c\_button\_\_container\_theme\_\_uEpvR ']"));  
 public static final Target *BTN\_SHOW\_CONTENT* = Target.*the*("Boton de mostrar contenido")  
 .locatedBy("//button[text()='Mostrar contenido bloqueado']");  
 public static final Target *IMG\_LINK\_URL* = Target.*the*("Click en la imagen de ingresa aqui")  
 .located(By.*xpath*("(//\*[@data-imagetype])[2]"));  
 public static final Target *OK\_BUTTON* = Target.*the*("Click al botón Entendido")  
 .locatedBy("((//div[text()='Entendido'])/..)[3]");  
 public static final Target *CONTINUE\_BUTTON* = Target.*the*("Click al botón Entendido")  
 .locatedBy("(//div[text()='Continuar']/..)");  
  
 public static final Target *DOCUMENT\_TYPE\_DROPDOWN* = Target.*the*("Desplegable tipo de documento")  
 .located(By.*id*("sltDocumentType"));  
 public static final Target *CLICK\_OPTION\_TYPE\_DROPDOWN* = Target.*the*("Clic en la opción {0}")  
 .located(By.*xpath*("//div[@id='dropdown\_options\_sltDocumentType']//li[@data-keyprop='{0}']"));  
  
 public static final Target *MENSAJE\_SUPERASTESINTENTOS* = Target.*the*("Mensaje por usuario incorrecto")  
 .locatedBy("//p[contains(text(),'Superaste el número de intentos.')]");  
  
 public static final Target *MENSAJE\_BLOQUEO* = Target.*the*("Mensaje por usuario incorrecto")  
 .locatedBy("//div[contains(text(),'Por tu seguridad, hemos bloqueado temporalmente el ingreso a tu cuenta.')]");  
  
 public static final Target *BUTTON\_CAMBIARCONTRASEÑA* = Target.*the*("clic en botón cambiar contraseña")  
 .locatedBy("//div[text()='Cambiar contraseña']/parent::button");  
  
}

package ui;  
  
import net.serenitybdd.screenplay.targets.Target;  
  
public class TransferFormPage {  
  
 public static final Target *INPUT\_DNI* = Target.*the*("escribir DNI")  
 .locatedBy("//input[@data-testid='inpDocumentNumber']");  
  
 public static final Target *INPUT\_CORREO* = Target.*the*("escribir correo")  
 .locatedBy("//input[@data-testid='inpEmail']");  
  
 public static final Target *INPUT\_CONFIRMAR\_CORREO* = Target.*the*("escribir confirmar correo")  
 .locatedBy("//input[@data-testid='inpRepeatEmail']");  
  
 public static final Target *INPUT\_CELULAR* = Target.*the*("escribir celular")  
 .locatedBy("//input[@data-testid='inpMobilephoneNumber']");  
  
 public static final Target *CHECK\_INFO* = Target.*the*("selecciona el check")  
 .locatedBy("//p[contains(text(),'Para recibir mejores oportunidades de ahorro e inversión, autorizo los ')]/parent::span");  
  
 public static final Target *BUTTON\_CONTINUAR* = Target.*the*("click en el botón continuar")  
 .locatedBy("//button//div[text()='Continuar']/..");  
  
 public static final Target *TEXT\_VALIDAR* = Target.*the*("validar texto al iniciar transferir AFP Integra")  
 .locatedBy("//div[contains(@class,'i-body') and contains(text(),'{0}')]");  
  
  
}

package ui;  
  
import net.serenitybdd.screenplay.targets.Target;  
  
public class TransferPage {  
  
 public static final Target *CAMBIARMEAQUI\_CABECERA* = Target.*the*("clic en botón cambiarme aquí cabecera")  
 .locatedBy("//a[@id='cambiarme\_aqui\_header']");  
  
 public static final Target *CAMBIARMEAQUI\_BANNER* = Target.*the*("clic en botón cambiarme aquí banner")  
 .locatedBy("//a[@id='cambiarme\_aqui\_segundo']");  
  
 public static final Target *CAMBIARMEAQUI\_CARRUSEL* = Target.*the*("clic en botón cambiarme aquí carrusel")  
 .locatedBy("//a[@id='cambiarme\_aqui\_tercero']");  
  
 public static final Target *CAMBIARMEAQUI\_CONTENEDOR* = Target.*the*("clic en botón cambiarme aquí contenedor")  
 .locatedBy("(//div[text()='Cambiarme aquí'])[4]/parent::a");  
  
 public static final Target *CAMBIARMEAQUI\_PIE* = Target.*the*("clic en botón cambiarme aquí pie de página")  
 .locatedBy("(//div[text()='Cambiarme aquí'])[6]/parent::a");  
  
 public static final Target *VALIDAR\_SOLICITUD* = Target.*the*("valida pantalla inicio de solicitud")  
 .locatedBy("(//p[contains(text(), '¡Empecemos con tu cambio a AFP Integra!')])[1]");  
  
  
}

package ui;  
  
import net.serenitybdd.screenplay.targets.Target;  
import org.openqa.selenium.By;  
  
public class VoluntaryContributionsPage {  
 public static final Target *LETS\_BEGIN\_BUTTON* = Target.*the*("Empecemos")  
 .locatedBy("//div[text()='Empecemos']/parent::button");  
  
 public static final Target *SELECT\_VOLUNTARYCONTRIBUTIONTYPE\_BUTTON* = Target.*the*("Selecciona el tipo {0}")  
 .locatedBy("//div[contains(@class,'i-hide-mobile')]//p[text()='{0}']/parent::div/following-sibling::button");  
  
 public static final Target *CONTINUE\_BUTTON* = Target.*the*("Continuar")  
 .locatedBy("//div[text()='Continuar']/parent::button");  
  
 public static final Target *CHECKBOX\_TYPEFUND* = Target.*the*("Tipo de fondo {0}")  
 .locatedBy("//h3[contains(text(),'{0}')]");  
  
 public static final Target *BUTTON\_OPENACCOUNT* = Target.*the*("Abrir Cuenta")  
 .locatedBy("//div[text()='Abrir Cuenta']/parent::button");  
  
 public static final Target *ACCOUNTCREATED\_LABEL* = Target.*the*("Cuenta creada")  
 .locatedBy("(//p[text()='Cuenta creada'])[1]");  
  
 public static final Target *MAKE\_A\_DEPOSIT\_BUTTON* = Target.*the*("Realizar deposito")  
 .locatedBy("(//div[text()='Realizar depósito']/parent::button)[1]");  
  
 public static final Target *DEPOSIT\_TYPE\_RADIOBUTTON* = Target.*the*("Tipo de deposito {0}")  
 .locatedBy("//div[text()='{0}']/parent::div/parent::label");  
  
 public static final Target *BANK\_RADIOBUTTON* = Target.*the*("Selecciona el banco {0}")  
 .locatedBy("//input[@type='radio' and @value='{0}']/parent::label");  
  
 public static final Target *AMOUNT\_INPUT* = Target.*the*("Monto")  
 .located(By.*id*("inputAmount"));  
  
 public static final Target *CONTINUE\_BUTTON\_2* = Target.*the*("Continuar")  
 .locatedBy("//div[text()='Continuar']/parent::button");  
  
 public static final Target *COUNTRY\_SELECT* = Target.*the*("País")  
 .located(By.*id*("sltCountry"));  
  
 public static final Target *IDENTITYTRIBUTTE\_INPUT* = Target.*the*("Número de identificación tributaria (NIT)")  
 .located(By.*id*("inpIdentityTributte"));  
  
 public static final Target *COUNTRYOPERATIONS\_SELECT* = Target.*the*("País de operaciones")  
 .located(By.*id*("sltCountryOperations"));  
  
 public static final Target *AMERICAN\_NATIONALITY\_CHECK* = Target.*the*("Check nacionalidad estadounidense")  
 .locatedBy("//span[contains(text(),'Tienes nacionalidad estadounidense o eres ciudadano(a)')]");  
  
 public static final Target *ID\_NATIONALITY\_EEUU\_INPUT* = Target.*the*("Número de documento fiscal de EEUU")  
 .located(By.*id*("inpHasNationalityEEUU"));  
  
 public static final Target *IS\_INVESTIGATED\_CHECK* = Target.*the*("Check ha sido investigado o procesado")  
 .locatedBy("//span[contains(text(),'Ha sido investigado o procesado')]");  
  
 public static final Target *INVESTIGATED\_DETAILS\_INPUT* = Target.*the*("Detallar delito y estado de la investigación")  
 .located(By.*id*("inpIsInvestigated"));  
  
 public static final Target *CONFIDENTIAL\_INFO\_CHECK* = Target.*the*("Check acepta información es CONFIDENCIAL")  
 .locatedBy("//span[contains(text(),'Acepto que la información brindada tiene carácter de Declaración Jurada CONFIDENCIAL')]");  
  
 public static final Target *MORE\_INFO\_CHECK* = Target.*the*("Check recibir más información")  
 .locatedBy("//span[contains(text(),'Me gustaría recibir información sobre otras oportunidades')]");  
  
 public static final Target *CONTINUE\_BUTTON\_3* = Target.*the*("Continuar")  
 .locatedBy("//div[text()='Continuar']/parent::button");  
  
 public static final Target *GENERATE\_TICKET\_BUTTON* = Target.*the*("Generar ticket")  
 .locatedBy("//div[text()='Generar ticket']/parent::button");  
  
 public static final Target *GENERATED\_YOUR\_TICKET\_LABEL* = Target.*the*("¡Listo! Hemos generado tu ticket")  
 .locatedBy("//p[text()='¡Listo! Hemos generado tu ticket']");  
  
 public static final Target *UNDERSTOOD\_BUTTON* = Target.*the*("Entendido")  
 .locatedBy("//div[text()='Entiendo']/parent::button");  
  
 public static final Target *COUNTRY\_ITEM* = Target.*the*("País {0}")  
 .locatedBy("//div[@id='dropdown\_options\_sltCountry']//li[text()='{0}']");  
  
 public static final Target *COUNTRYOPERATIONS\_ITEM* = Target.*the*("País de operaciones {0}")  
 .locatedBy("//div[@id='dropdown\_options\_sltCountryOperations']//li[text()='{0}']");  
  
 public static final Target *ORIGIN\_SELECT* = Target.*the*("Origen del dinero desplegable número {0}")  
 .locatedBy("(//input[@id='sltOrigin'])[{0}]");  
  
 public static final Target *ORIGINOPTIONS\_ITEM* = Target.*the*("Tipo de origen {0} para para el desplegable número {1}")  
 .locatedBy("(//div[@id='dropdown\_options\_sltOrigin']//li[text()='{0}'])[{1}]");  
  
 public static final Target *ADD\_OTHER\_BUTTON* = Target.*the*("Agregar otro origen del dinero")  
 .locatedBy("//div[text()='Agregar otro']/parent::button");  
  
 public static final Target *BANK\_INPUT* = Target.*the*("Entidad financiera {0}")  
 .locatedBy("(//input[@id='inpEntity'])[{0}]");  
  
}

package ui;  
  
import net.serenitybdd.core.annotations.findby.By;  
import net.serenitybdd.screenplay.targets.Target;  
  
public class WithdrawVoluntaryContributionsPage {  
  
 public static final Target *CHKBUTTON\_WITHDRAWTYPE* = Target.*the*("Tipo de retiro {0}")  
 .locatedBy("//p[text()='Selecciona el tipo de retiro']/following-sibling::div//div[text()='{0}']");  
  
 public static final Target *AMOUNT\_WITHDRAWTYPE* = Target.*the*("Monto retiro")  
 .located(By.*id*("inpAmount"));  
  
 public static final Target *CHECKBOX\_BANK* = Target.*the*("Banco {0}")  
 .locatedBy("//input[@name='radioBank' and @value='{0}']/..");  
  
 public static final Target *LABEL\_TYPEACCOUNT* = Target.*the*("Tipo de cuenta")  
 .locatedBy("//input[@id='sltTypeAccount']/parent::div");  
  
 public static final Target *SELECT\_TYPEACCOUNT* = Target.*the*("Seleccionar Tipo de cuenta")  
 .locatedBy("//ul/li[text()='{0}']");  
  
 public static final Target *INPUT\_ACCOUNTNUMBER* = Target.*the*("Número de cuenta")  
 .located(By.*id*("inpBankAccount"));  
  
 public static final Target *INPUT\_CCINUMBER* = Target.*the*("Número CCI")  
 .located(By.*id*("inpCci"));  
  
 public static final Target *CHECK\_TERMS\_AND\_CONDITIONS* = Target.*the*("Check terminos y condiciones")  
 .locatedBy("//p[contains(text(),'He leído y acepto los ')]");  
  
 public static final Target *BUTTON\_CONTINUAR* = Target.*the*("Continuar")  
 .locatedBy("(//div[text()='Continuar']/parent::button)[1]");  
  
 public static final Target *BUTTON\_CONTINUAR\_2* = Target.*the*("Continuar")  
 .locatedBy("(//div[text()='Continuar']/parent::button)[2]");  
  
 public static final Target *CONFIRM\_WITHDRAW\_BUTTON* = Target.*the*("Confirmar retiro")  
 .locatedBy("//button//div[text()='Confirmar retiro']/..");  
  
 public static final Target *LABEL\_ERROR\_MESSAGE* = Target.*the*("Mensaje de error {0}")  
 .locatedBy("(//span[contains(@class,'error') and text()='{0}'])[1]");  
  
 public static final Target *LABEL\_SECURITYCODE\_ERROR\_MESSAGE* = Target.*the*("Mensaje de error de código de seguridad {0}")  
 .locatedBy("//p[contains(text(),'{0}')]");  
  
 public static final Target *BUTTON\_ENTENDIDO* = Target.*the*("Click en boton entendido")  
 .locatedBy("//div[@class='Modal\_c\_modal\_\_VTsr3 Modal\_c\_modal\_\_IsOpen\_\_irDTM ']/descendant::button[@class='Button\_c\_button\_\_I\_x2\_ Button\_c\_button\_\_container\_theme\_\_uEpvR ']");  
  
 public static final Target *LABEL\_MESSAGE\_SALDOMINIMO* = Target.*the*("Mensaje informativo {0}")  
 .locatedBy("//div[contains(@class,'i-text-center') and text()='{0}']");  
  
  
  
}

RUNNER

package com.sura.web.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.sura.web.stepdefinitions"},  
 plugin = {"pretty", "json:target/cucumber-reports/cucumber.json"},  
 tags = "@changeFund\_esc03",  
 snippets = CucumberOptions.SnippetType.*CAMELCASE*)  
  
public class CucumberTestSuite {  
  
}

STEP DEFINITIONS

package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.en.And;  
import io.cucumber.java.en.Given;  
import io.cucumber.java.en.Then;  
import net.serenitybdd.screenplay.conditions.Check;  
import questions.ChangeFundQuestions;  
import tasks.changeFund.\*;  
import tasks.common.ClickElement;  
import tasks.common.ClickElementJs;  
import tasks.common.WaitFor;  
import tasks.login.PrincipalUrl;  
import tasks.outlook.OpenOutlookAndRetrieveCode;  
import ui.ChangeFundPage;  
import ui.CookiesPage;  
  
import static net.serenitybdd.screenplay.GivenWhenThen.*seeThat*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorCalled*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
import static org.hamcrest.Matchers.*containsString*;  
  
public class ChangeFundStepDefinitions {  
  
 @And("el usuario hace clic en el botón Mis trámites del menu")  
 public void elUsuarioIngresaElNombreDeUsuario() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonMyProcedures());  
 }  
  
 @And("el usuario hace clic en el botón Cambia tu tipo de fondo")  
 public void elUsuarioHaceClicEnElBotónCambiaTuTipoDeFondo() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonChangeTypeFund());  
 }  
  
 @Then("el usuario se encuentra en la página de consideraciones")  
 public void elUsuarioSeEncuentraEnLaPáginaDeConsideraciones() {  
 }  
  
 @And("el usaurio hace clic en el botón Sí, empecemos")  
 public void elUsaurioHaceClicEnElBotónSíEmpecemos() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonYesLetStart());  
 }  
  
 @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));  
 }  
  
 @And("el usuario hace clic en el botón Continuar")  
 public void elUsuarioHafeClicEnElBotónContinuar() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonContinue());  
 }  
  
 @Then("el usuario debería ver {string} en la página")  
 public void elUsuarioDeberíaVerEnLaPágina(String text) {  
 *theActorInTheSpotlight*().should(*seeThat*(ChangeFundQuestions.*ValidateMessage*(text), *containsString*(text)));  
 }  
  
 @And("el usuario acepta los términos y condiciones")  
 public void elUsuarioAceptaLosTérminosYCondiciones() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(ChangeFundPage.*ACCEPT\_TERMS\_CHECKBOX*, true));  
 }  
  
 @And("el usuario ingresa el código de seguridad obtenido")  
 public void elUsuarioIngresaElCódigoDeSeguridadObtenido() {  
 String securityCode = *theActorInTheSpotlight*().recall("securityCode");  
 *theActorInTheSpotlight*().attemptsTo(EnterSecurityCodeInFields.*with*(securityCode));  
 }  
  
 @And("el usuario hace clic en el botón Continuar para enviar código de seguridad")  
 public void elUsuarioHaceClicEnElBotónContinuarParaEnviarCódigoDeSeguridad() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonContinueToCode());  
 }  
  
 @And("el usuario hacel clic en Validar código")  
 public void elUsuarioHacelClicEnValidarCódigo() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonValidateCode());  
 }  
  
 @And("el usuario ingresa a outlook con su correo: {string} y clave: {string}, búsca el úmtimo correo con el asunto {string} y obtiene el código de seguridad")  
 public void elUsuarioIngresaAOutlookConSuCorreoYClaveBúscaElÚmtimoCorreoConElAsuntoYObtieneElCódigoDeSeguridad(String correo, String clave, String asunto) {  
 *theActorInTheSpotlight*().attemptsTo(OpenOutlookAndRetrieveCode.*withCredentials*(correo, clave, asunto));  
 }  
  
 @Then("el usuario valida el mensaje {string}")  
 public void elUsuarioValidaElMensaje(String expectedMessage) {  
 *theActorInTheSpotlight*().should(  
 *seeThat*(ChangeFundQuestions.*ValidateMessageChangeFund*(expectedMessage), *containsString*(expectedMessage)));  
 }  
  
 @Given("que el usuario está en la página principal")  
 public void queElUsuarioEstáEnLaPáginaPrincipal() {  
 *theActorCalled*("user").wasAbleTo(PrincipalUrl.*CambioAFP*());  
 }  
  
 @And("el usuario acepta el cookis")  
 public void elUsuarioAceptaElCookis() {  
 *theActorInTheSpotlight*().attemptsTo(WaitFor.*seconds*(10));  
 *theActorInTheSpotlight*().attemptsTo(Check.*whether*(CookiesPage.*COOKIES\_ACCEPT*.resolveFor(  
 *theActorInTheSpotlight*()).isVisible()).andIfSo(ClickElement.*on*(CookiesPage.*COOKIES\_ACCEPT*, true)));  
 }  
  
 @And("el usuario hace clic en la opción Cambiar tu Tipo de Fondo")  
 public void elUsuarioHaceClicEnLaOpciónCambiarTuTipoDeFondo() {  
 *theActorInTheSpotlight*().attemptsTo(WaitFor.*seconds*(3));  
 *theActorInTheSpotlight*().attemptsTo(ClickElementJs.*on*(ChangeFundPage.*BUTTON\_CAMBIAR\_FONDO*));  
 }  
  
 @And("el usuario hace clic en el botón iniciar trámite")  
 public void elUsuarioHaceClicEnElBotónIniciarTrámite() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(ChangeFundPage.*BUTTON\_INICIAR\_TRAMITE*, true));  
  
 }  
}

package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.Before;  
import net.serenitybdd.model.environment.ConfiguredEnvironment;  
import net.serenitybdd.screenplay.abilities.BrowseTheWeb;  
import net.serenitybdd.screenplay.actors.Cast;  
import net.serenitybdd.screenplay.actors.OnStage;  
import net.thucydides.model.util.EnvironmentVariables;  
import org.openqa.selenium.WebDriver;  
import static drivers.AWSDeviceFarmDriverFactory.*createRemoteDriver*;  
import static drivers.LocalDriverFactory.*createLocalDriver*;  
  
public class Hooks {  
 private static final EnvironmentVariables *environmentVariables* = ConfiguredEnvironment.*getEnvironmentVariables*();  
 private static final String *BROWSER* = *environmentVariables*.getProperty("browser");  
  
 @Before  
 public void setTheStage() throws Exception {  
 WebDriver driver;  
 String executionEnv = System.*getenv*("EXECUTION\_ENV");  
 if ("AWS".equalsIgnoreCase(executionEnv)) {  
 driver = *createRemoteDriver*(*BROWSER*);  
 } else {  
 driver = *createLocalDriver*(*BROWSER*);  
 }  
 OnStage.*setTheStage*(Cast.*whereEveryoneCan*(BrowseTheWeb.*with*(driver)));  
 }  
}

package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.en.And;  
import io.cucumber.java.en.Given;  
import io.cucumber.java.en.Then;  
import io.cucumber.java.en.When;  
import questions.ElementVisible;  
import questions.LoginQuestions;  
import tasks.login.\*;  
import ui.LoginPage;  
  
import static net.serenitybdd.screenplay.GivenWhenThen.*seeThat*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorCalled*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
import static org.hamcrest.Matchers.*containsString*;  
import static org.hamcrest.Matchers.*equalTo*;  
import static org.hamcrest.Matchers.*anyOf*;  
import static org.hamcrest.Matchers.*is*;  
  
public class LoginStepDefinitions {  
  
 @Given("que el usuario está en la página de inicio de sesión")  
 public void queElUsuarioEstáEnLaPáginaDeInicioDeSesión() {  
 *theActorCalled*("user").wasAbleTo(OpenUrl.*loginAFP*());  
 }  
  
 @When("el usuario selecciona el tipo de documento {string}")  
 public void elUsuarioSeleccionaElTipoDeDocumento(String typeDocument) {  
 *theActorInTheSpotlight*().attemptsTo(SelectDocumentType.*withData*(typeDocument));  
 }  
  
 @And("el usuario ingresa el nombre de usuario {string}")  
 public void elUsuarioIngresaElNombreDeUsuario(String user) {  
 *theActorInTheSpotlight*().attemptsTo(InputUser.*withData*(user));  
 }  
  
 @And("el usuario ingresa la contraseña {string}")  
 public void elUsuarioIngresaLaContraseña(String password) {  
 *theActorInTheSpotlight*().attemptsTo(InputPassword.*withData*(password));  
 }  
  
 @And("el usuario hace clic en el botón de inicio de sesión")  
 public void elUsuarioHaceClicEnElBotónDeInicioDeSesión() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonEnter());  
 }  
  
 @Then("el usuario debería ver el mensaje de error {string}")  
 public void elUsuarioDeberíaVerElMensajeDeError(String message) {  
 *theActorInTheSpotlight*().should(*seeThat*(LoginQuestions.*ValidateMessage*(), *equalTo*(message)));  
 }  
  
 @Then("el usuario debería ver el mensaje de campo user incompleto {string}")  
 public void elUsuarioDeberíaVerElMensajeDeCampoUserInvalido(String message) {  
 *theActorInTheSpotlight*().should(*seeThat*(LoginQuestions.*ValidateMessageFieldUser*(), *containsString*(message)));  
 }  
  
 @Then("el usuario debería ver el mensaje de campo user inválido {string}")  
 public void elUsuarioDeberíaVerElMensajeDeCampoUserInválido(String message) {  
 *theActorInTheSpotlight*().should(*seeThat*(LoginQuestions.*ValidateMessageModalUser*(), *equalTo*(message)));  
 }  
  
 @Then("el usuario debería ver el mensaje de campo password incompleta {string}")  
 public void elUsuarioDeberíaVerElMensajeDeCampoPasswordIncompleta(String message) {  
 *theActorInTheSpotlight*().should(*seeThat*(LoginQuestions.*ValidateMessageFieldPassword*(), *equalTo*(message)));  
 }  
  
  
 @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")  
 )  
 )  
 );  
 }  
  
 @Then("el usuario debe ver mensaje de contraseña inválida")  
 public void elUsuarioDebeVerMensajeDeContraseñaInválida() {  
 *theActorInTheSpotlight*().should(  
 *seeThat*(  
 "El mensaje de bienvenida es correcto",  
 LoginQuestions.*ValidateMessagedos*(),  
 *anyOf*(  
 *containsString*("Tienes un (1) intento más para ingresar tu contraseña."),  
 *containsString*("Tienes dos (2) intentos más para ingresar tu contraseña.")  
 )  
 )  
 );  
  
  
 }  
  
 @Then("el usuario debería ver el mensaje de datos incorrectos o no es cliente de AFP Integra")  
 public void elUsuarioDeberíaVerElMensajeDeDatosIncorrectosONoEsClienteDeAFPIntegra() {  
  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 LoginPage.*VALIDAR\_MENSAJE\_DATOSINCORRECT*), *is*(true)));  
 }  
  
}

package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.en.And;  
import tasks.dashboard.CloseModalPresent;  
  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
  
public class OutlookStepDefinitions {  
  
 @And("el usuario cierra va ventana modal informativa")  
 public void el\_usuario\_cierra\_va\_ventana\_modal\_informativa() {  
 *theActorInTheSpotlight*().attemptsTo(CloseModalPresent.*close*());  
 }  
  
}

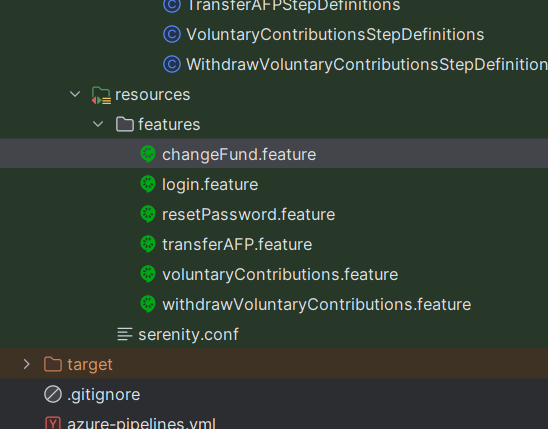
package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.en.And;  
import io.cucumber.java.en.Then;  
import io.cucumber.java.en.When;  
import questions.ElementVisible;  
import questions.ResetPasswordQuestions;  
import tasks.common.ClickElement;  
import tasks.resetPassword.\*;  
import ui.ResetPasswordPage;  
import static org.hamcrest.Matchers.*is*;  
import static net.serenitybdd.screenplay.GivenWhenThen.*seeThat*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
import static org.hamcrest.Matchers.*containsString*;  
  
public class ResetPasswordStepDefinitions {  
  
 @When("el usuario selecciona la opción de ¿Olvidaste tu contraseña?")  
 public void elUsuarioSeleccionaLaOpciónDe() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonResetPassword());  
 }  
  
 @And("el usuario envía el mensaje de verificación al correo electrónico")  
 public void elUsuarioHaceClicEnElBotónDeContinuar() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonNext());  
 }  
  
 @And("el usuario hace click en el boton {string}")  
 public void elUsuarioHaceClickEnElBoton(String text) {  
 *theActorInTheSpotlight*().attemptsTo(ButtonSendEmail.*withData*(text));  
 }  
  
 @Then("el usuario debería ver el mensaje de confirmación {string}")  
 public void elUsuarioDeberíaVerElMensajeDeConfirmación(String message) {  
 *theActorInTheSpotlight*().should(*seeThat*(ResetPasswordQuestions.*ValidateMessageConfirmation*(), *containsString*(message.trim())));  
 }  
  
 @When("el usuario abre el enlace en su correo")  
 public void elUsuarioAbreElEnlaceEnSuCorreo() {  
 *theActorInTheSpotlight*().wasAbleTo(OpenUrlOutlook.*loginOutlook*());  
 }  
  
 @And("el usuario ingresa el usuario {string} del correo")  
 public void elUsuarioIngresaElUsuarioDelCorreo(String userEmail) {  
 *theActorInTheSpotlight*().attemptsTo(InputUserEmail.*withData*(userEmail));  
 }  
  
 @And("el usuario hace clic en el siguiente botón")  
 public void elUsuarioHaceClicEnElSiguienteBotón() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonNext());  
 }  
  
 @And("el usuario ingresa la contraseña {string} del correo")  
 public void elUsuarioIngresaLaContraseñaDelCorreo(String passwordEmail) {  
 *theActorInTheSpotlight*().attemptsTo(InputPasswordEmail.*withData*(passwordEmail));  
 }  
  
 @And("el usuario {string} desea recordar la sesión")  
 public void elUsuarioDeseaRecordarLaSesión(String opcion) {  
 *theActorInTheSpotlight*().attemptsTo(ButtonYesNotSession.*withData*(opcion));  
 }  
  
 @And("el usuario busca el correo {string}")  
 public void elUsuarioBuscaElCorreo(String titleEmail) {  
 *theActorInTheSpotlight*().attemptsTo(InputSearchEmail.*withData*(titleEmail));  
 // theActorInTheSpotlight().attemptsTo(new ButtonSearch());  
 *theActorInTheSpotlight*().attemptsTo(new SelectFirstEmail());  
 *theActorInTheSpotlight*().attemptsTo(ButtonShowContent.*now*());  
 *theActorInTheSpotlight*().attemptsTo(ClickLinkChoose.*now*());  
 *theActorInTheSpotlight*().attemptsTo(SwitchToTab.*to*(2));  
 *theActorInTheSpotlight*().attemptsTo(new ButtonOk());  
 }  
  
 @And("el usuario ingresa una nueva contraseña {string}")  
 public void elUsuarioIngresaUnaNuevaContraseña(String newPassword) {  
 *theActorInTheSpotlight*().attemptsTo(InputNewPassword.*withData*(newPassword));  
 }  
  
 @Then("el usuario debería ver el mensaje de verificación {string}")  
 public void elUsuarioDeberíaVerElMensajeDeVerificación(String messageVerificationEmail) {  
 // Get the message  
 String message = ResetPasswordQuestions.*ValidateMessageConfirmationEmail*().answeredBy(*theActorInTheSpotlight*());  
  
 // Print the message to the console  
 System.*out*.println("Confirmation Email Message: " + message);  
  
 // Store the message in a variable  
 String confirmationMessage = message;  
 *theActorInTheSpotlight*().should(*seeThat*(ResetPasswordQuestions.*ValidateMessageConfirmationEmail*(), *containsString*(messageVerificationEmail)));  
 *theActorInTheSpotlight*().attemptsTo(new ButtonUnderStood());  
 }  
  
 @And("el usuario hace clic en el siguiente")  
 public void elUsuarioHaceClicEnElSiguiente() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonContinue());  
 }  
  
  
  
  
  
 @Then("el usuario debería ver el mensaje de Superaste el número de intentos")  
 public void elUsuarioDeberíaVerElMensajeDeSuperasteElNúmeroDeIntentos() {  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 ResetPasswordPage.*MENSAJE\_SUPERASTESINTENTOS*), *is*(true)));  
 }  
  
 @Then("el usuario visualiza el mensaje de bloqueo temporal")  
 public void elUsuarioVisualizaElMensajeDeBloqueoTemporal() {  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 ResetPasswordPage.*MENSAJE\_BLOQUEO*), *is*(true)));  
 }  
  
 @And("el usuario hace clic en el botón Cambiar contraseña")  
 public void elUsuarioHaceClicEnElBotónCambiarContraseña() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(ResetPasswordPage.*BUTTON\_CAMBIARCONTRASEÑA*,true));  
 }  
}

package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.en.And;  
import io.cucumber.java.en.Then;  
import io.cucumber.java.en.When;  
import net.serenitybdd.screenplay.targets.Target;  
import questions.ElementVisible;  
import tasks.common.ClickElement;  
import tasks.common.EnterText;  
import tasks.common.ScrollTo;  
import tasks.common.WaitFor;  
import ui.HomePage;  
import ui.TransferFormPage;  
import ui.TransferPage;  
  
  
import static org.hamcrest.Matchers.*is*;  
import static net.serenitybdd.screenplay.GivenWhenThen.*seeThat*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
import static tasks.common.ScrollTo.*element*;  
  
import net.serenitybdd.screenplay.abilities.BrowseTheWeb;  
import org.openqa.selenium.WebDriver;  
  
  
public class TransferAFPStepDefinitions {  
  
 @When("el usuario hace clic en el botón de Hazte cliente")  
 public void elUsuarioHaceClicEnElBotónDeHazteCliente() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(HomePage.*BUTTON\_HAZTE\_CLIENTE*, true));  
 }  
  
  
 @And("el usuario hace clic en el botón Cámbiate Ya")  
 public void elUsuarioHaceClicEnElBotónCámbiateYa() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(HomePage.*BUTTON\_CAMBIATE\_YA*, true));  
  
 }  
  
 @And("el usuario hace clic en el botón Cámbiarme aquí {string}")  
 public void elUsuarioHaceClicEnElBotónCámbiarmeAquí(String Posicion) {  
 // Obtener el WebDriver asociado al actor  
 WebDriver driver = BrowseTheWeb.*as*(*theActorInTheSpotlight*()).getDriver();  
  
 // Guardar el identificador de la ventana actual  
 String currentWindow = driver.getWindowHandle();  
  
 // Hacer clic en el botón según la posición  
 switch (Posicion.toLowerCase()) {  
 case "de la cabecera":  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferPage.*CAMBIARMEAQUI\_CABECERA*, true));  
 break;  
 case "del banner":  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferPage.*CAMBIARMEAQUI\_BANNER*, true));  
 break;  
 case "del carrusel vertical":  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferPage.*CAMBIARMEAQUI\_CARRUSEL*, true));  
 break;  
 case "del contenedor azul":  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferPage.*CAMBIARMEAQUI\_CONTENEDOR*, true));  
 break;  
 case "del pie de página":  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferPage.*CAMBIARMEAQUI\_PIE*, true));  
 break;  
 default:  
 throw new IllegalArgumentException("No se reconoce el botón: " + Posicion);  
 }  
  
 // Cambiar a la nueva ventana y cerrar la anterior  
 driver.getWindowHandles()  
 .stream()  
 .filter(window -> !window.equals(currentWindow)) // Filtrar para obtener la nueva ventana  
 .findFirst()  
 .ifPresent(newWindow -> {  
 driver.switchTo().window(currentWindow).close(); // Cerrar la ventana actual  
 driver.switchTo().window(newWindow); // Cambiar a la nueva ventana  
 });  
  
 }  
  
 @Then("el usuario valida que ingreso al formulario de solicitud de traspaso")  
 public void elUsuarioValidaQueIngresoAlFormularioDeSolicitudDeTraspaso() {  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(TransferPage.*VALIDAR\_SOLICITUD*),*is*(true)));  
 }  
  
 @And("el usuario ingresa {string} en DNI")  
 public void elUsuarioIngresaEnDNI(String DNI) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_DNI*,  
 DNI.trim()));  
 }  
  
 @And("el usuario ingresa {string} en Correo Electrónico")  
 public void elUsuarioIngresaEnCorreoElectrónico(String correo) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_CORREO*,  
 correo.trim()));  
 }  
  
 @And("el usuario ingresa {string} en Confirmar Correo Electrónico")  
 public void elUsuarioIngresaEnConfirmarCorreoElectrónico(String correo) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_CONFIRMAR\_CORREO*,  
 correo.trim()));  
 }  
  
 @And("el usuario ingresa {string} en Celular")  
 public void elUsuarioIngresaEnCelular(String celular) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(TransferFormPage.*INPUT\_CELULAR*,  
 celular.trim()));  
 }  
  
 @And("el usuario {string} el check de mejores oportunidades de ahorro e inversión")  
 public void elUsuarioElCheckDeMejoresOportunidadesDeAhorroEInversión(String marca) {  
 if (marca.trim().equalsIgnoreCase("marca")){  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferFormPage.*CHECK\_INFO*,false));  
 }  
 }  
  
  
 @And("el usuario sigue el proceso haciendo clic en el botón Continuar")  
 public void elUsuarioSigueElProcesoHaciendoClicEnElBotónContinuar() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(TransferFormPage.*BUTTON\_CONTINUAR*, true));  
  
 }  
  
 @Then("el usuario valida el mensaje {string} al realizar un traspaso AFP Integra")  
 public void elUsuarioValidaElMensajeAlRealizarUnTraspasoAFPIntegra(String mensaje) {  
 // theActorInTheSpotlight().attemptsTo(ScrollTo.element(TransferFormPage.TEXT\_VALIDAR));  
 *theActorInTheSpotlight*().attemptsTo(WaitFor.*seconds*(15)); //tiempo explicito  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 TransferFormPage.*TEXT\_VALIDAR*, mensaje), *is*(true)));  
 }  
}

package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.en.And;  
import io.cucumber.java.en.Then;  
import questions.ElementVisible;  
import tasks.common.ClickElement;  
import tasks.common.EnterText;  
import tasks.voluntaryContributions.SelectOriginOfMoney;  
import ui.DashboardPage;  
import ui.VoluntaryContributionsPage;  
import ui.WithdrawVoluntaryContributionsPage;  
  
import static net.serenitybdd.screenplay.GivenWhenThen.*seeThat*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
import static org.hamcrest.Matchers.*is*;  
  
public class VoluntaryContributionsStepDefinitions {  
  
 @And("el usuario selecciona la opción Abre o deposita a tu cuenta de Aportes Voluntarios")  
 public void elusuarioseleccionalaopciónAbreodepositaatucuentadeAportesVoluntarios() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 DashboardPage.*MY\_PROCEDURES\_OPTIONS\_BUTTON*,  
 false,  
 "Abre o deposita a tu cuenta de Aportes Voluntarios"));  
 }  
  
 @And("el usuario hace clic a Empecemos")  
 public void elUsuarioHaceClicAEmpecemos() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*LETS\_BEGIN\_BUTTON*,  
 false));  
 }  
  
 @And("el usuario selecciona Abrir Una Cuenta {string}")  
 public void elUsuarioSeleccionaAbrirUnaCuenta(String tipo\_cuenta) {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*SELECT\_VOLUNTARYCONTRIBUTIONTYPE\_BUTTON*,  
 false,  
 tipo\_cuenta));  
 }  
  
 @And("el usuario hace clic en Continuar")  
 public void elUsuarioHaceClicEnContinuar() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*CONTINUE\_BUTTON*,  
 false  
 ));  
 }  
  
 @And("el usuario elige el tipo de fondo {string}")  
 public void elUsuarioEligeElTipoDeFondo(String tipo\_fondo) {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*CHECKBOX\_TYPEFUND*,  
 true,  
 tipo\_fondo));  
 }  
  
 @And("el usuario hace clic en Abrir Cuenta")  
 public void elUsuarioHaceClicEnAbrirCuenta() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*BUTTON\_OPENACCOUNT*,  
 true  
 ));  
 }  
  
 @And("el usuario hace clic en Realizar depósito")  
 public void elUsuarioHaceClicEnRealizarDepósito() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*MAKE\_A\_DEPOSIT\_BUTTON*,  
 false  
 ));  
 }  
  
 @Then("el usuario valida que la cuenta se haya creado correctamente")  
 public void elUsuarioValidaQueLaCuentaSeHayaCreadoCorrectamente() {  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 VoluntaryContributionsPage.*ACCOUNTCREATED\_LABEL*), *is*(true)));  
 }  
  
 @And("el usuario elige el tipo de deposito {string}")  
 public void elUsuarioEligeElTipoDeDeposito(String tipo\_deposito) {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*DEPOSIT\_TYPE\_RADIOBUTTON*,  
 false,  
 tipo\_deposito  
 ));  
 }  
  
 @And("el usuario elige el banco {string}")  
 public void elUsuarioEligeElBanco(String banco) {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*BANK\_RADIOBUTTON*,  
 false,  
 banco  
 ));  
 }  
  
 @And("el usuario ingresa el monto que desea aportar de {string} soles")  
 public void elUsuarioIngresaElMontoQueDeseaAportarDeSoles(String monto) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(  
 VoluntaryContributionsPage.*AMOUNT\_INPUT*,  
 monto.trim()  
 ));  
 }  
  
 @And("el usuario hace clic en Continuar para depositar")  
 public void elUsuarioHaceClicEnContinuarParaDepositar() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 VoluntaryContributionsPage.*CONTINUE\_BUTTON\_2*,  
 false  
 ));  
 }  
  
  
 @And("el usuario selecciona {string} como país donde vive actualmente y el número de nit {string} si se requiera")  
 public void elUsuarioSeleccionaComoPaísDondeViveActualmenteYElNúmeroDeNitSiSeRequiera(String pais, String nit) {  
  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*COUNTRY\_SELECT*,false));  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*COUNTRY\_ITEM*,false, pais));  
  
 if (!nit.trim().isEmpty()){  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(VoluntaryContributionsPage.*IDENTITYTRIBUTTE\_INPUT*, nit.trim()));  
 }  
  
 }  
  
 @And("el usuario selecciona {string} como país de origen del dinero que usará para sus operaciones")  
 public void elUsuarioSeleccionaComoPaísDeOrigenDelDineroQueUsaráParaSusOperaciones(String pais) {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*COUNTRYOPERATIONS\_SELECT*,false));  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*COUNTRYOPERATIONS\_ITEM*,false, pais));  
 }  
  
 @And("el usuario selecciona {string} como origen del dinero")  
 public void elUsuarioSeleccionaComoOrigenDelDinero(String valor\_origen) {  
 *theActorInTheSpotlight*().attemptsTo(SelectOriginOfMoney.*withValues*(valor\_origen));  
 }  
  
 @And("el usuario marca el check de nacionalidad americana si tiene documento fiscal de EEUU {string}")  
 public void elUsuarioMarcaElCheckDeNacionalidadAmericanaSiTieneDocumentoFiscalDeEEUU(String documento\_eeuu) {  
 if (!documento\_eeuu.trim().isEmpty()){  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*AMERICAN\_NATIONALITY\_CHECK*,false));  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(VoluntaryContributionsPage.*ID\_NATIONALITY\_EEUU\_INPUT*, documento\_eeuu));  
 }  
 }  
  
 @And("el usuario marca el check de Ha sido investigado si tiene algún delito y lo detalla {string}")  
 public void elUsuarioMarcaElCheckDeHaSidoInvestigadoSiTieneAlgúnDelitoYLoDetalla(String delitos) {  
 if (!delitos.trim().isEmpty()){  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*IS\_INVESTIGATED\_CHECK*,false));  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(VoluntaryContributionsPage.*INVESTIGATED\_DETAILS\_INPUT*, delitos));  
 }  
 }  
  
 @And("el usuario marca el check de Acepto que la información es Confidencial {string}")  
 public void elUsuarioMarcaElCheckDeAceptoQueLaInformaciónEsConfidencial(String acepta\_info\_conf) {  
 if (acepta\_info\_conf.trim().equalsIgnoreCase("SI")){  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*CONFIDENTIAL\_INFO\_CHECK*,false));  
 }  
 }  
  
 @And("el usuario marca {string} en el check de Me gustaría recibir más información sobre oportunidades de ahorro")  
 public void elUsuarioMarcaEnElCheckDeMeGustaríaRecibirMásInformaciónSobreOportunidadesDeAhorro(String mas\_informacion) {  
 if (mas\_informacion.trim().equalsIgnoreCase("SI")){  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(VoluntaryContributionsPage.*MORE\_INFO\_CHECK*,false));  
 }  
 }  
}

package com.sura.web.stepdefinitions;  
  
import io.cucumber.java.en.And;  
import io.cucumber.java.en.Then;  
import io.cucumber.java.en.When;  
import model.Bank;  
import questions.ElementVisible;  
import tasks.changeFund.EnterSecurityCodeInFields;  
import tasks.common.ClickElement;  
import tasks.common.EnterText;  
import tasks.dashboard.MyProceduresOptions;  
import tasks.outlook.LoginOutlook;  
import tasks.outlook.SearchSecurityCode;  
import tasks.withdrawVoluntaryContributions.ButtonEntendido;  
import tasks.withdrawVoluntaryContributions.InputAmountWithdraw;  
import tasks.withdrawVoluntaryContributions.SelectWithdrawType;  
import ui.WithdrawVoluntaryContributionsPage;  
  
import static net.serenitybdd.screenplay.GivenWhenThen.*seeThat*;  
import static net.serenitybdd.screenplay.actors.OnStage.*theActorInTheSpotlight*;  
import static org.hamcrest.Matchers.*is*;  
  
public class WithdrawVoluntaryContributionsStepDefinitions {  
  
 @And("el usuario selecciona la opción Retira tus Aportes Voluntarios")  
 public void elUsuarioDeseaRecordarLaSesión() {  
 *theActorInTheSpotlight*().attemptsTo(MyProceduresOptions.*click*("Retira tus Aportes "));  
 }  
  
 @And("el usuario selecciona el tipo de retiro {string}")  
 public void elUsuarioSeleccionaElTipoDeRetiro(String tipo\_retiro) {  
 *theActorInTheSpotlight*().attemptsTo(SelectWithdrawType.*click*(tipo\_retiro));  
 }  
  
 @And("el usuario ingresa el monto de {string} soles")  
 public void elUsuarioIngresaElMontoDeSoles(String amount) {  
 *theActorInTheSpotlight*().attemptsTo(InputAmountWithdraw.*withAmount*(amount));  
 }  
  
 @And("el usuario selecciona el banco {string} donde recibirá su dinero")  
 public void elUsuarioSeleccionaElBancoDondeRecibiráSuDinero(String bank) {  
 bank = Bank.*fromString*(bank).getCode();  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(  
 WithdrawVoluntaryContributionsPage.*CHECKBOX\_BANK*,  
 true,  
 bank));  
 }  
  
 @And("el usuario selecciona el tipo de cuenta {string}")  
 public void elUsuarioSeleccionaElTipoDeCuenta(String tipo\_cuenta) {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(WithdrawVoluntaryContributionsPage.*LABEL\_TYPEACCOUNT*,true));  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(WithdrawVoluntaryContributionsPage.*SELECT\_TYPEACCOUNT*, true,  
 tipo\_cuenta));  
 }  
  
 @And("el usuario ingresa {string} en número de cuenta")  
 public void elUsuarioIngresaEnNúmeroDeCuenta(String numero\_cuenta) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(WithdrawVoluntaryContributionsPage.*INPUT\_ACCOUNTNUMBER*,  
 numero\_cuenta.trim()));  
 }  
  
 @And("el usuario ingresa {string} en CCI")  
 public void elUsuarioIngresaEnCCI(String cci) {  
 if (cci != null && !cci.trim().isEmpty()) {  
 *theActorInTheSpotlight*().attemptsTo(EnterText.*into*(WithdrawVoluntaryContributionsPage.*INPUT\_CCINUMBER*,  
 cci.trim()));  
 }  
 }  
  
 @Then("el usuario valida el resumen de la solicitud")  
 public void elUsuarioValidaElResumenDeLaSolicitud() {  
 }  
  
 @And("el usuario acepta los términos y condiciones del retiro")  
 public void elUsuarioAceptaLosTérminosYCondicionesDelRetiro() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(WithdrawVoluntaryContributionsPage.*CHECK\_TERMS\_AND\_CONDITIONS*, true));  
 }  
  
 @And("el usuario hace clic en el botón Continar retiro")  
 public void elUsuarioHaceClicEnElBotónContinarRetiro() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(WithdrawVoluntaryContributionsPage.*BUTTON\_CONTINUAR*, true));  
 }  
  
 @And("el usuario hace clic en el botón Continuar para enviar código de seguridad del retiro")  
 public void elUsuarioHaceClicEnElBotónContinuarParaEnviarCódigoDeSeguridadDelRetiro() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(WithdrawVoluntaryContributionsPage.*BUTTON\_CONTINUAR\_2*, true));  
 }  
  
 @And("el usuario ingresa a outlook con su correo: {string} y clave: {string}")  
 public void elUsuarioIngresaAOutlookConSuCorreoYClave(String usuario, String clave) {  
 *theActorInTheSpotlight*().attemptsTo(LoginOutlook.*login*(usuario,clave));  
 }  
  
 @And("el usuario búsca el último correo con el asunto {string} y obtiene el código de seguridad")  
 public void elUsuarioBúscaElÚltimoCorreoConElAsuntoYObtieneElCódigoDeSeguridad(String asunto) {  
 *theActorInTheSpotlight*().attemptsTo(SearchSecurityCode.*With*(asunto));  
 }  
  
 @When("el usuario hace clic en Confirmar Retiro")  
 public void elUsuarioHaceClicEnConfirmarRetiro() {  
 *theActorInTheSpotlight*().attemptsTo(ClickElement.*on*(WithdrawVoluntaryContributionsPage.*CONFIRM\_WITHDRAW\_BUTTON*, true));  
 }  
  
 @Then("el usuario valida mensaje de retiro de aportes voluntario exitoso")  
 public void elUsuarioValidaMensajeDeRetiroDeAportesVoluntarioExitoso() {  
 }  
  
 @Then("el usuario visualizará el mensaje {string}")  
 public void elUsuarioVisualizaráElMensaje(String mensaje) {  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 WithdrawVoluntaryContributionsPage.*LABEL\_ERROR\_MESSAGE*, mensaje), *is*(true)));  
 }  
  
 @And("el usuario ingresa un código de seguridad {string}")  
 public void elUsuarioIngresaUnCódigoDeSeguridad(String codigo) {  
 *theActorInTheSpotlight*().attemptsTo(EnterSecurityCodeInFields.*with*(codigo));  
 }  
  
 @Then("el usuario visualiza el mensaje de error: {string}")  
 public void elUsuarioVisualizaElMensajeDeError(String mensaje) {  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 WithdrawVoluntaryContributionsPage.*LABEL\_SECURITYCODE\_ERROR\_MESSAGE*, mensaje), *is*(true)));  
 }  
  
 @And("el usuario hace clic en el boton Entendido")  
 public void elUsuarioHaceClicEnElBotonEntendido() {  
 *theActorInTheSpotlight*().attemptsTo(new ButtonEntendido());  
 }  
  
 @Then("el usuario visualizará el mensaje informativo {string}")  
 public void elUsuarioVisualizaráElMensajeInformativo(String mensajeRetiro) {  
 *theActorInTheSpotlight*().should(*seeThat*(ElementVisible.*isVisible*(  
 WithdrawVoluntaryContributionsPage.*LABEL\_MESSAGE\_SALDOMINIMO*, mensajeRetiro), *is*(true)));  
 }  
  
  
}

FEATURES



@changeFund @web  
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  
  
 @changeFund\_esc01  
 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 su correo: "<correo>" y clave: "<clave>", búsca el úmtimo correo con el asunto "AFP Integra - Código de Seguridad para Cambio de Fondo" y obtiene el código de seguridad  
 And el usuario ingresa el código de seguridad obtenido  
 And el usuario hacel clic en Validar código  
 *# 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) | 7mbKzo0b7V |  
*# pendiente agregar mas casos*  
  
@changeFund\_esc02  
 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*  
  
*##nuevo cambio de fondo alternativo cambio de fondo desde página principal*  
@changeFund\_esc03  
 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 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 su correo: "<correo>" y clave: "<clave>", búsca el úmtimo correo con el asunto "AFP Integra - Código de Seguridad para Cambio de Fondo" y obtiene el código de seguridad*  
*# And el usuario ingresa el código de seguridad obtenido*  
*# And el usuario hacel clic en Validar código*  
*# # 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 |  
*# pendiente agregar mas casos*

@inicioSesion @web  
Feature: Iniciar sesión en la aplicación web  
 Como usuario de la aplicación web  
 Quiero iniciar sesión con mis credenciales válidas  
 Para acceder a mi Agencia Digital  
  
 @esc01 @login @happy  
 Scenario Outline: "<Caso de Prueba>" El usuario inicia sesión con tipo documento "<Tipo Documento>" con credenciales válidas  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<Tipo Documento>"  
 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  
  
 Examples:  
 | Caso de Prueba | Tipo Documento | Usuario | Contraseña |  
 | CP01 | DNI | 72159283 | Cercania2024 |  
 | CP02 | CE | 001108718 | Cercania2024 |  
 | CP03 | PAS | D588394 | Cercania2024x |  
 | CP04 | PTP | 85808405 | Cercania2024 |  
*# | CP05 | CRE | 09783628 | Cercania2024 |*  
*# | CP06 | CIE | 0962710232 | Cercania2024 |*  
*# | CP07 | CSR | 742402050 | Cercania2024 |*  
*# | CP08 | CPP | 1234567890 | Cercania2024 |*  
*# | CP09 | CIA | 72777884 | Cercania2024 |*  
*# | CP10 | CFA | 31974330 | Cercania2024 |*  
  
@unhappy @login @esc02  
 Scenario Outline: "<Caso de Prueba>" El usuario no puede iniciar sesión por el usuario incompleto  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<Tipo Documento>"  
 And el usuario ingresa el nombre de usuario "<Usuario>"  
 Then el usuario debería ver el mensaje de campo user incompleto "No es un campo valido"  
  
 Examples:  
 | Caso de Prueba | Tipo Documento | Usuario |  
 | CP11 | DNI | 71 |  
  
 @unhappy @login @esc03  
 Scenario Outline: "<Caso de Prueba>" El usuario no puede iniciar sesión porque el usuario es inválido  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<Tipo Documento>"  
 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  
 Then el usuario debería ver el mensaje de campo user inválido "Puede que no hayas ingresado correctamente tus datos o que no seas cliente de AFP Integra."  
  
 Examples:  
 | Caso de Prueba | Tipo Documento | Usuario | Contraseña |  
 | CP12 | CE | 001108710 | Cercania2021 |  
  
 @unhappy @login @esc04  
 Scenario Outline: "<Caso de Prueba>" El usuario no puede iniciar sesión porque la contraseña es incompleta  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<Tipo Documento>"  
 And el usuario ingresa el nombre de usuario "<Usuario>"  
 And el usuario ingresa la contraseña "<Contraseña>"  
 Then el usuario debería ver el mensaje de campo password incompleta "Debe ingresar mínimo 8 dígitos"  
  
 Examples:  
 | Caso de Prueba | Tipo Documento | Usuario | Contraseña |  
 | CP13 | PAS | D588394 | Cercan |  
  
 @unhappy @login @esc05  
 Scenario Outline: "<Caso de Prueba>" El usuario no puede iniciar sesión porque la contraseña es inválida  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<Tipo Documento>"  
 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  
 Then el usuario debe ver mensaje de contraseña inválida  
  
 Examples:  
 | Caso de Prueba | Tipo Documento | Usuario | Contraseña |  
 *# | CP14 | CIA | 72777884 | Cercania2020 |*  
| CP14 | DNI | 45661984 | Cercania2020 |  
  
 @unhappy @login @esc06  
 Scenario Outline: "<Caso de prueba>" El usuario no se encuentra registrado  
 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 "abchg5g55"  
 And el usuario hace clic en el botón de inicio de sesión  
 Then el usuario debería ver el mensaje de datos incorrectos o no es cliente de AFP Integra  
  
 Examples:  
 | Caso de prueba | tipoDocumento | usuario |  
 | CP13 | DNI | 99999999 |

@resetPassword  
Feature: Cambiar contraseña y validar inicio de sesión en la aplicación web  
 Como usuario de la aplicación web  
 Quiero cambiar mi contraseña y validar el inicio de sesión  
 Para asegurarme de que puedo acceder a mi cuenta de manera segura  
  
 @esc01resetPassword @happy  
 Scenario Outline: "<Caso de prueba>" El usuario con tipo de documento "<Tipo de Documento>" cambia la contraseña y valida el inicio de sesión  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona la opción de ¿Olvidaste tu contraseña?  
 And el usuario selecciona el tipo de documento "<Tipo de Documento>"  
 And el usuario ingresa el nombre de usuario "<Usuario>"  
 And el usuario hace clic en el siguiente botón  
 And el usuario hace click en el boton "Enviar correo"  
 Then el usuario debería ver el mensaje de confirmación "Te hemos enviado un correo para generar tu contraseña al siguiente correo:"  
 When el usuario abre el enlace en su correo  
 And el usuario ingresa el usuario "[pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe)" del correo  
 And el usuario hace clic en el siguiente botón  
 And el usuario ingresa la contraseña "8mbKzo0b7V" del correo  
 And el usuario hace clic en el siguiente botón  
 And el usuario "Sí" desea recordar la sesión  
 And el usuario busca el correo "AFP Integra - Generación de contraseña"  
 And el usuario ingresa una nueva contraseña "<Nueva Contraseña>"  
 And el usuario hace clic en el siguiente  
 Then el usuario debería ver el mensaje de verificación "¡Estupendo!"  
 When el usuario selecciona el tipo de documento "<Tipo de Documento>"  
 And el usuario ingresa el nombre de usuario "<Usuario>"  
 And el usuario ingresa la contraseña "<Nueva Contraseña>"  
 And el usuario hace clic en el botón de inicio de sesión  
 Then el usuario valida el mensaje de bienvenida  
  
 Examples:  
 | Caso de prueba | Tipo de Documento | Usuario | Nueva Contraseña |  
 *# | CP01 | DNI | 00797231 | Cercania2024 |*  
 *# | CP02 | CE | 001108718 | Cercania2024 |*  
*# | CP03 | PAS | D588394 | Cercania2024 |*  
*# | CP04 | PTP | 85808405 | Cercania2024 |*  
 *# | CP05 | CRE | 09783628 | Cercania2024 |*  
 *# | CP06 | CIE | 0962710232 | Cercania2024 |*  
*# | CP07 | CSR | 742402050 | Cercania2024 |*  
*# | CP08 | CPP | 1234567890 | Cercania2024 |*  
 *# | CP09 | CIA | 72777884 | Cercania2024 |*  
 *# | CP10 | CFA | 31974330 | Cercania2024 |*  
| CPXX | DNI | 21535926 | Cercania2024 |  
 @esc02resetPassword @unhappy  
 Scenario Outline: "<Caso de prueba>" El usuario no puede cambiar la contraseña por usuario incompleto  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona la opción de ¿Olvidaste tu contraseña?  
 And el usuario selecciona el tipo de documento "<tipoDocumento>"  
 And el usuario ingresa el nombre de usuario "<usuario>"  
 Then el usuario debería ver el mensaje de campo user incompleto "No es un campo valido"  
  
 Examples:  
 | Caso de prueba | tipoDocumento | usuario |  
 | CP11 | DNI | 71 |  
  
 @esc03resetPassword @unhappy  
 Scenario Outline: "<Caso de prueba>" El usuario no puede cambiar la contraseña porque el usuario es inválido  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona la opción de ¿Olvidaste tu contraseña?  
 And el usuario selecciona el tipo de documento "<tipoDocumento>"  
 And el usuario ingresa el nombre de usuario "<usuario>"  
 And el usuario hace clic en el siguiente botón  
 Then el usuario debería ver el mensaje de campo user inválido "Puede que no hayas ingresado correctamente tus datos o que no seas cliente de AFP Integra."  
  
 Examples:  
 | Caso de prueba | tipoDocumento | usuario |  
 | CP12 | CE | 001108710 |  
  
 *#falta el escenario 4 y 5 implemetar FALTA*  
@esc04resetPassword @happy  
 Scenario Outline: "<Caso de prueba>" El usuario no puede cambiar la contraseña porque el correo no está registrado  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona la opción de ¿Olvidaste tu contraseña?  
 And el usuario selecciona el tipo de documento "<tipoDocumento>"  
 And el usuario ingresa el nombre de usuario "<usuario>"  
 Then el usuario debería ver el mensaje de correo no registrado  
*##CONSULTAR A HEBER*  
Examples:  
 | Caso de prueba | tipoDocumento | usuario |  
 | CP13 | DNI | 07701658 |  
  
  
 @esc05resetPassword @happy *##(revisar)*  
Scenario Outline: "<Caso de prueba>" El usuario se bloquea al 3er intento por ingresar usuario incorrecto y debe cambiar contraseña  
 Given que el usuario está en la página de inicio de sesión  
 And el usuario selecciona el tipo de documento "<Tipo de Documento>"  
 And el usuario ingresa el nombre de usuario "<Usuario>"  
 And el usuario ingresa la contraseña "abchg5g55"  
 And el usuario hace clic en el botón de inicio de sesión  
 And el usuario hace clic en el botón de inicio de sesión  
 And el usuario hace clic en el botón de inicio de sesión  
 Then el usuario debería ver el mensaje de Superaste el número de intentos  
 And el usuario hace clic en el botón de inicio de sesión  
 Then el usuario visualiza el mensaje de bloqueo temporal  
 And el usuario hace clic en el botón Cambiar contraseña  
 And el usuario hace click en el boton "Enviar correo"  
 Then el usuario debería ver el mensaje de confirmación "Te hemos enviado un correo para generar tu contraseña al siguiente correo:"  
 When el usuario abre el enlace en su correo  
 And el usuario ingresa el usuario "[pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe)" del correo  
 And el usuario hace clic en el siguiente botón  
 And el usuario ingresa la contraseña "8mbKzo0b7V" del correo  
 And el usuario hace clic en el siguiente botón  
 And el usuario "Sí" desea recordar la sesión  
 And el usuario busca el correo "AFP Integra - Generación de contraseña"  
 And el usuario ingresa una nueva contraseña "<Nueva Contraseña>"  
 And el usuario hace clic en el siguiente  
 Then el usuario debería ver el mensaje de verificación "¡Estupendo!"  
 When el usuario selecciona el tipo de documento "<Tipo de Documento>"  
 And el usuario ingresa el nombre de usuario "<Usuario>"  
 And el usuario ingresa la contraseña "<Nueva Contraseña>"  
 And el usuario hace clic en el botón de inicio de sesión  
 Then el usuario valida el mensaje de bienvenida  
  
 Examples:  
 | Caso de prueba | Tipo de Documento | Usuario | Nueva Contraseña |  
 | CP13 | DNI | 21535926 | Cercania2024 |  
 *# | CP13 | DNI | 06197878 | Cercania2024 |*  
 *# | CP13 | DNI | 04413667 | Cercania2024 |*  
 *# | CP13 | DNI | 72159283 | Cercania2024 |*  
  
 *#Superaste el número de intentos. Para acceder nuevamente haz clic en "Olvidé mi contraseña"32956445.*

@trasferAFP  
Feature: Cambio de fondos de pensiones AFP  
 Como usuario de un fondo de pensión AFP  
 Quiero realizar el cambio de fondo de pensión a AFP Integra  
 Para poder obtener mejores beneficios  
*#escenario ejecucion exitosa*  
@transferAFP\_esc01 @happy  
 Scenario Outline: "<Caso de Prueba>" El usuario valida que el botón "<Posición>" redireccione a la Solicitud de Traspaso a AFP Integra  
 Given que el usuario está en la página principal  
 And el usuario acepta el cookis  
 When el usuario hace clic en el botón de Hazte cliente  
 And el usuario hace clic en el botón Cámbiate Ya  
 And el usuario hace clic en el botón Cámbiarme aquí "<Posición>"  
 Then el usuario valida que ingreso al formulario de solicitud de traspaso  
  
 Examples:  
 | Caso de Prueba | Posición |  
 | CP01 | De la Cabecera |  
 | CP02 | Del Banner |  
 | CP03 | Del Carrusel vertical |  
 | CP04 | Del Contenedor Azul |  
 | CP05 | Del Pie de página |  
  
*##separar por escenarios para el lunes*  
@transferAFP\_esc02 @happy  
 Scenario Outline: <Caso de Prueba> El usuario valida el mensaje "<mensaje>" al realizar un traspaso AFP Integra  
 Given que el usuario está en la página principal  
 And el usuario acepta el cookis  
 When el usuario hace clic en el botón de Hazte cliente  
 And el usuario hace clic en el botón Cámbiate Ya  
 And el usuario hace clic en el botón Cámbiarme aquí "De la Cabecera"  
 And el usuario ingresa "<DNI>" en DNI  
 And el usuario ingresa "<correo>" en Correo Electrónico  
 And el usuario ingresa "<correo>" en Confirmar Correo Electrónico  
 And el usuario ingresa "<celular>" en Celular  
 And el usuario "marca" el check de mejores oportunidades de ahorro e inversión  
 And el usuario sigue el proceso haciendo clic en el botón Continuar  
 Then el usuario valida el mensaje "<mensaje>" al realizar un traspaso AFP Integra  
  
 Examples:  
 | Caso de Prueba | DNI | correo | celular | mensaje |  
 | CP06 | 45335136 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Tú ya eres parte de AFP Integra |  
 | CP07 | 99999999 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Actualmente no te encuentras en una AFP |  
 | CP08 | 06936146 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Actualmente no te encuentras en una AFP |  
 | CP09 | 75953952 | [irvin.mendoza20mr@gmail.com](mailto:irvin.mendoza20mr@gmail.com) | 929214484 | Actualmente no te encuentras en una AFP |  
 | CP10 | 44545454 | [proveedorirvin.mendoza@sura.pe](mailto:proveedorirvin.mendoza@sura.pe) | 929214450 | Cuéntanos, ¿actualmente vives en la dirección que aparece en tu DNI? |  
  
  
 *#DNI ya esta registrado en AFP integra*  
 *#DNI que no esta registrado en reniec*  
 *#usar DNI nuevo ,no esta en AFP, ONP y te deja afiliar*  
 *#usar DNI,esta ONP y te deja afiliar*  
  
*#usar DNI,esta en AFP diferente a integra y mensaje*  
  
  
  
@transferAFP\_esc03 @happy  
 Scenario Outline: "<Caso de Prueba>" El usuario ya se encuentra registrado o usuario no se encuentra en una AFP  
 Given que el usuario está en la página principal  
 And el usuario acepta el cookis  
 When el usuario hace clic en el botón de Hazte cliente  
 And el usuario hace clic en el botón Cámbiate Ya  
 And el usuario hace clic en el botón Cámbiarme aquí "De la Cabecera"  
 And el usuario ingresa Número de DNI "<DNI>"  
 And el usuario ingresa Correo Electrónico "<correo>"  
 And el usuario confirma Correo Electrónico "<correo>"  
 And el usuario ingresa número de Celular "<celular>"  
 *##no marca y marca*  
And el usuario "marca" el check de mejores oportunidades de ahorro e inversión  
 And el usuario hace clic en botón Continuar  
 Then el usuario valida el mensaje "<mensajeTraspaso>" de informativo  
  
 Examples:  
 | Caso de Prueba | DNI | celular | mensajeTraspaso |  
 | CP01 | 45335136 | 929214484 | usuario ya registrado |  
 | CP01 | 99999999 | 929214484 | usuario no tiene AFP |  
  
  
 @tranferAFP\_esc04 @happy  
 Scenario: "<Caso de Prueba>" El usuario realiza una solicitud de traspaso a AFP Integra  
 Given que el usuario está en la página principal  
 When el usuario hace clic en el botón de Hazte cliente  
 And el usuario hace clic en el botón Cámbiate Ya  
 And el usuario hace clic en el botón Cámbiarme aquí  
 And el usuario ingresa Número de DNI "<DNI>"  
 And el usuario ingresa Correo Electrónico "<correo>"  
 And el usuario ingresa número de Celular "<celular>"  
 And el usuario marca el check de mejores oportunidades de ahorro e inversión  
 And el usuario hace clic en botón Continuar  
 And el usuario ingresa el nombre de usuario "<usuario>"  
 Then el usuario valida el mensaje de bienvenida

@AV  
Feature: Aportes Voluntarios  
 Como un usuario registrado  
 Quiero abrir una cuenta de aportes voluntarios y realizar depósitos  
 Para gestión de ahorros  
  
 @AV\_SFP  
 Scenario Outline: "<Caso de Prueba>" El usuario abre una cuenta de aportes voluntarios sin fin previsional de tipo "<tipo fondo>"  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Abre o deposita a tu cuenta de Aportes Voluntarios  
 And el usuario hace clic a Empecemos  
 And el usuario selecciona Abrir Una Cuenta "Con Aportes Voluntarios Sin Fin Previsional"  
 And el usuario hace clic en Continuar  
 And el usuario elige el tipo de fondo "<tipo fondo>"  
 When el usuario hace clic en Abrir Cuenta  
 Then el usuario valida que la cuenta se haya creado correctamente  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | tipo fondo |  
 | CP01 | CIE | 0962710232 | Cercania2024 | Fondo 1 |  
 | CP02 | DNI | 40038791 | Cercania2024 | Fondo 2 |  
  
 *#agregar mas casos fondo 3 - 1 caso mas*  
  
@AV\_CFP  
 Scenario Outline: "<Caso de Prueba>" El usuario abre una cuenta de aportes voluntarios con fin previsional de tipo "<tipo fondo>"  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Abre o deposita a tu cuenta de Aportes Voluntarios  
 And el usuario hace clic a Empecemos  
 And el usuario selecciona Abrir Una Cuenta "Con Aportes Voluntarios Con Fin Previsional"  
 And el usuario hace clic en Continuar  
 And el usuario elige el tipo de fondo "<tipo fondo>"  
 When el usuario hace clic en Abrir Cuenta  
 Then el usuario valida que la cuenta se haya creado correctamente  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | tipo fondo |  
 | CP03 | CIA | 72777884 | Cercania2024 | Fondo 3 |  
  
  
 *#agregar mas casos 2 casos mas*  
  
  
@AV\_SFP\_DEPOSITO  
 Scenario Outline: "<Caso de Prueba>" El usuario abre una cuenta de aportes voluntarios SFP de tipo "<tipo fondo>" y realiza un deposito único con la entidad bancaria "<banco>"  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Abre o deposita a tu cuenta de Aportes Voluntarios  
 And el usuario hace clic a Empecemos  
 And el usuario selecciona Abrir Una Cuenta "Con Aportes Voluntarios Sin Fin Previsional"  
 And el usuario hace clic en Continuar  
 And el usuario elige el tipo de fondo "<tipo fondo>"  
 When el usuario hace clic en Abrir Cuenta  
 Then el usuario valida que la cuenta se haya creado correctamente  
 And el usuario hace clic en Realizar depósito  
 And el usuario elige el tipo de deposito "Depósito único"  
 *# And el usuario elige el tipo de deposito "Depósito recurrente"*  
And el usuario elige el banco "<banco>"  
 And el usuario ingresa el monto que desea aportar de "<monto>" soles  
 And el usuario hace clic en Continuar para depositar  
 And el usuario selecciona "<pais>" como país donde vive actualmente y el número de nit "<nit>" si se requiera  
 And el usuario selecciona "<pais operaciones>" como país de origen del dinero que usará para sus operaciones  
 And el usuario selecciona "<origen del dinero>" como origen del dinero  
 And el usuario marca el check de nacionalidad americana si tiene documento fiscal de EEUU "<documento fiscal EEUU>"  
 And el usuario marca el check de Ha sido investigado si tiene algún delito y lo detalla "<detalle delito>"  
 And el usuario marca el check de Acepto que la información es Confidencial "<acepta info confidencial>"  
 And el usuario marca "<acepta mas info>" en el check de Me gustaría recibir más información sobre oportunidades de ahorro  
  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | tipo fondo | banco | monto | pais | nit | pais operaciones | origen del dinero | documento fiscal EEUU | detalle delito | acepta info confidencial | acepta mas info |  
 *# |CP04| CIA | 72777884 | Cercania2024 | Fondo 3 | Yape | 100 | PERÚ | | PERÚ | Otros (especificar) - BBVA, Prestación de servicios - bcp, Venta de propiedad - interbank | | | | |*  
 *# |CP05| CIA | 72777884 | Cercania2024 | Fondo 3 | BCP | 100 | PERÚ | | PERÚ | Otros (especificar) - BBVA, Prestación de servicios - bcp, Venta de propiedad - interbank | | | | |*  
| CP06 | CIA | 72777884 | Cercania2024 | Fondo 3 | Interbank | 100 | ZAMBIA | 9999 | CHILE | Otros (especificar) - BBVA, Prestación de servicios - bcp, Venta de propiedad - interbank | 999999 | delito | Si | Si |  
*# |CP07| CIA | 72777884 | Cercania2024 | Fondo 3 | Scotiabank | 100 | PERÚ | | PERÚ | Otros (especificar) - BBVA, Prestación de servicios - bcp, Venta de propiedad - interbank | | | | |*  
*# |CP08| CIA | 72777884 | Cercania2024 | Fondo 3 | BBVA | 100 | PERÚ | | PERÚ | Otros (especificar) - BBVA, Prestación de servicios - bcp, Venta de propiedad - interbank | | | | |*

@wyvc  
Feature: Retiro de Aportes Voluntarios  
 Como usuario afiliado  
 Quiero realizar el retiro de mis aportes voluntarios de manera segura y eficiente  
 Para poder disponer de mis fondos según lo requiera.  
  
 @wyvc\_esc01 @happy  
 Scenario Outline: "<Caso de Prueba>" El usuario realiza un retiro de aportes voluntarios parcial seleccionando banco "<banco>" y tipo "<tipo de cuenta>"  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Retira tus Aportes Voluntarios  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario selecciona el tipo de retiro "Retiro parcial"  
 And el usuario ingresa el monto de "<monto>" soles  
 And el usuario selecciona el banco "<banco>" donde recibirá su dinero  
 And el usuario selecciona el tipo de cuenta "<tipo de cuenta>"  
 And el usuario ingresa "<cuenta>" en número de cuenta  
 And el usuario ingresa "<cci>" en CCI  
 And el usuario acepta los términos y condiciones del retiro  
 And el usuario hace clic en el botón Continar retiro  
 And el usuario hace clic en el botón Continuar para enviar código de seguridad del retiro  
 And el usuario ingresa a outlook con su correo: "<correo>" y clave: "<clave>"  
 And el usuario búsca el último correo con el asunto "AFP Integra - Código de Seguridad para Retiro de Aporte Voluntario" y obtiene el código de seguridad  
 And el usuario ingresa el código de seguridad obtenido  
 When el usuario hace clic en Confirmar Retiro  
 Then el usuario valida mensaje de retiro de aportes voluntario exitoso  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | monto | banco | tipo de cuenta | cuenta | cci | correo | clave |  
 | CP01 | DNI | 04413667 | Cercania2024 | 11 | BBVA | Ahorro Soles | 0011 - 1234 - 0212345678 | 123-123-123123123213-21 | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP02 | DNI | 04413667 | Cercania2024 | 2487.87 | BBVA | Ahorro Soles | 0011 - 1234 - 0212345678 | 123-123-123123123213-21 | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP03 | DNI | 04413667 | Cercania2024 | 10.1 | BBVA | Cuenta Corriente Soles | 0011-0661-0100004571 | 123-123-123123123213-21 | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP04 | DNI | 21535926 | Cercania2024 | 2487.87 | BBVA | Cuenta Corriente Soles | 0011-0661-0100004571 | 123-123-123123123213-21 | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP05 | DNI | 21535926 | Cercania2024 | 10.1 | BCP | Ahorro Soles | 191-00709940-0-95 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP06 | DNI | 21535926 | Cercania2024 | 2487.87 | BCP | Ahorro Soles | 191-00709940-0-95 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP07 | DNI | 72159283 | Cercania2024 | 10.1 | BCP | Cuenta Corriente Soles | 191-0070994-0-09 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP08 | DNI | 72159283 | Cercania2024 | 2487.87 | BCP | Cuenta Corriente Soles | 191-0070994-0-09 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP09 | DNI | 72159283 | Cercania2024 | 10.1 | INTERBANK | Ahorro Soles | 100-0000137229 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP10 | DNI | 06197878 | Cercania2024 | 2487.87 | INTERBANK | Ahorro Soles | 100-0000137229 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP11 | DNI | 06197878 | Cercania2024 | 10.1 | INTERBANK | Cuenta Corriente Soles | 100-0000137229 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP12 | DNI | 06197878 | Cercania2024 | 2487.87 | INTERBANK | Cuenta Corriente Soles | 100-0000137229 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP13 | DNI | 32956445 | Cercania2024 | 10.1 | SCOTIABANK | Ahorro Soles | 010-0160000 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP14 | DNI | 32956445 | Cercania2024 | 2487.87 | SCOTIABANK | Ahorro Soles | 010-0160000 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP15 | DNI | 32956445 | Cercania2024 | 10.1 | SCOTIABANK | Cuenta Corriente Soles | 010-0160000 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP16 | DNI | 16002553 | Cercania2024 | 2487.87 | SCOTIABANK | Cuenta Corriente Soles | 010-0160000 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
  
 @wyvc\_esc02 @happy  
 Scenario Outline: "<Caso de Prueba>" El usuario realiza un retiro de aportes voluntarios total seleccionando banco "<banco>" y tipo "<tipo de cuenta>"  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Retira tus Aportes Voluntarios  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario selecciona el tipo de retiro "Retiro total"  
 And el usuario selecciona el banco "<banco>" donde recibirá su dinero  
 And el usuario selecciona el tipo de cuenta "<tipo de cuenta>"  
 And el usuario ingresa "<cuenta>" en número de cuenta  
 And el usuario ingresa "<cci>" en CCI  
 And el usuario acepta los términos y condiciones del retiro  
 And el usuario hace clic en el botón Continar retiro  
 And el usuario hace clic en el botón Continuar para enviar código de seguridad del retiro  
 And el usuario ingresa a outlook con su correo: "<correo>" y clave: "<clave>"  
 And el usuario búsca el último correo con el asunto "AFP Integra - Código de Seguridad para Retiro de Aporte Voluntario" y obtiene el código de seguridad  
 And el usuario ingresa el código de seguridad obtenido  
 When el usuario hace clic en Confirmar Retiro  
 Then el usuario valida mensaje de retiro de aportes voluntario exitoso  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | banco | tipo de cuenta | cuenta | cci | correo | clave |  
 | CP17 | DNI | 42694046 | Cercania2024 | BBVA | Ahorro Soles | 0011 - 1234 - 0212345678 | 123-123-123123123213-21 | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP18 | DNI | 41217929 | Cercania2024 | BBVA | Cuenta Corriente Soles | 0011-0661-0100004571 | 123-123-123123123213-21 | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP19 | DNI | 21434393 | Cercania2024 | BCP | Ahorro Soles | 191-00709940-0-95 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP20 | DNI | 80253182 | Cercania2024 | BCP | Cuenta Corriente Soles | 191-0070994-0-09 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP21 | DNI | 41244783 | Cercania2024 | INTERBANK | Ahorro Soles | 100-0000137229 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP22 | DNI | 32814303 | Cercania2024 | INTERBANK | Cuenta Corriente Soles | 100-0000137229 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP23 | DNI | 40337251 | Cercania2024 | SCOTIABANK | Ahorro Soles | 010-0160000 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
 | CP24 | DNI | 42821897 | Cercania2024 | SCOTIABANK | Cuenta Corriente Soles | 010-0160000 | | [pruebascalidad.arqti1@sura.pe](mailto:pruebascalidad.arqti1@sura.pe) | 8mbKzo0b7V |  
  
 @wyvc\_esc03 @unhappy  
 Scenario Outline: "<Caso de Prueba>" El usuario intenta retirar un monto mayor a lo permitido  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Retira tus Aportes Voluntarios  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario selecciona el tipo de retiro "Retiro parcial"  
 And el usuario ingresa el monto de "<monto>" soles  
 Then el usuario visualizará el mensaje "El monto de retiro excede al fondo actual"  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | monto |  
 | CP25 | DNI | 06540138 | Cercania2024 | 99999999 |  
  
 @wyvc\_esc04 @unhappy  
 Scenario Outline: "<Caso de Prueba>" El usuario intenta retirar un monto menor a lo permitido  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Retira tus Aportes Voluntarios  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario selecciona el tipo de retiro "Retiro parcial"  
 And el usuario ingresa el monto de "<monto>" soles  
 Then el usuario visualizará el mensaje "El monto mínimo de retiro es de 10.00"  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | monto |  
 | CP26 | DNI | 06540138 | Cercania2024 | 9.99 |  
  
 @wyvc\_esc05 @unhappy  
 Scenario Outline: "<Caso de Prueba>" El usuario ingresa el banco "<banco>" y tipo "<tipo de cuenta>" con un número de cuenta no válido  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Retira tus Aportes Voluntarios  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario selecciona el tipo de retiro "Retiro total"  
 And el usuario selecciona el banco "<banco>" donde recibirá su dinero  
 And el usuario selecciona el tipo de cuenta "<tipo de cuenta>"  
 And el usuario ingresa "<cuenta>" en número de cuenta  
 And el usuario ingresa "<cci>" en CCI  
 Then el usuario visualizará el mensaje "Número de cuenta incorrecto"  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | banco | tipo de cuenta | cuenta | cci |  
 | CP27 | DNI | 06540138 | Cercania2024 | BBVA | Ahorro Soles | 000000000000000000 | 0000000000000000000 |  
 | CP28 | DNI | 06540138 | Cercania2024 | BBVA | Cuenta Corriente Soles | 000000000000000001 | 0000000000000000000 |  
 | CP29 | DNI | 06540138 | Cercania2024 | INTERBANK | Ahorro Soles | 000000000000 | |  
 | CP30 | DNI | 06540138 | Cercania2024 | INTERBANK | Cuenta Corriente Soles | 000000000000 | |  
 | CP31 | DNI | 06540138 | Cercania2024 | BCP | Ahorro Soles | 0000000000000 | |  
 | CP32 | DNI | 06540138 | Cercania2024 | BCP | Cuenta Corriente Soles | 000000000000 | |  
 | CP33 | DNI | 06540138 | Cercania2024 | SCOTIABANK | Ahorro Soles | 000000000 | |  
 | CP34 | DNI | 06540138 | Cercania2024 | SCOTIABANK | Cuenta Corriente Soles | 000000000 | |  
  
 @wyvc\_esc06 @unhappy  
 Scenario Outline: "<Caso de Prueba>" El usuario ingresa el banco "<banco>" y tipo "<tipo de cuenta>" con un código de seguridad incorrecto  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Retira tus Aportes Voluntarios  
 And el usaurio hace clic en el botón Sí, empecemos  
 And el usuario selecciona el tipo de retiro "Retiro parcial"  
 And el usuario ingresa el monto de "<monto>" soles  
 And el usuario selecciona el banco "<banco>" donde recibirá su dinero  
 And el usuario selecciona el tipo de cuenta "<tipo de cuenta>"  
 And el usuario ingresa "<cuenta>" en número de cuenta  
 And el usuario ingresa "<cci>" en CCI  
 And el usuario acepta los términos y condiciones del retiro  
 And el usuario hace clic en el botón Continar retiro  
 And el usuario hace clic en el botón Continuar para enviar código de seguridad del retiro  
 And el usuario ingresa un código de seguridad "999999"  
 When el usuario hace clic en Confirmar Retiro  
 Then el usuario visualiza el mensaje de error: "Es posible que no estés ingresando el código correctamente"  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña | monto | banco | tipo de cuenta | cuenta | cci |  
 | CP35 | DNI | 42247066 | Cercania2024 | 100 | BCP | Ahorro Soles | 57997499829034 | |  
 | CP36 | DNI | 42247066 | Cercania2024 | 100 | BCP | Cuenta Corriente Soles | 191-0070994-0-09 | |  
 | CP37 | DNI | 42247066 | Cercania2024 | 100 | BBVA | Ahorro Soles | 0011 - 1234 - 0212345678 | 123-123-123123123213-21 |  
 | CP38 | DNI | 25731851 | Cercania2024 | 100 | BBVA | Cuenta Corriente Soles | 0011-0661-0100004571 | 123-123-123123123213-21 |  
 | CP39 | DNI | 25731851 | Cercania2024 | 100 | INTERBANK | Ahorro Soles | 100-0000137229 | |  
 | CP40 | DNI | 25731851 | Cercania2024 | 100 | INTERBANK | Cuenta Corriente Soles | 100-0000137229 | |  
 | CP41 | DNI | 40770739 | Cercania2024 | 100 | SCOTIABANK | Ahorro Soles | 010-0160000 | |  
 | CP42 | DNI | 40770739 | Cercania2024 | 100 | SCOTIABANK | Cuenta Corriente Soles | 010-0160000 | |  
  
 @wyvc\_esc07 @unhappy  
 Scenario Outline: "<Caso de Prueba>" El usuario intenta retirar sin saldo mínimo  
 Given que el usuario está en la página de inicio de sesión  
 When el usuario selecciona el tipo de documento "<tipo documento>"  
 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  
 Then el usuario valida el mensaje de bienvenida  
 And el usuario cierra va ventana modal informativa  
 And el usuario hace clic en el botón Mis trámites del menu  
 And el usuario selecciona la opción Retira tus Aportes Voluntarios  
 Then el usuario visualizará el mensaje informativo "Para realizar un retiro, debes tener un saldo mínimo de S/10.00 en tu cuenta"  
 And el usuario hace clic en el boton Entendido  
  
 Examples:  
 | Caso de Prueba | tipo documento | usuario | contraseña |  
 | CP43 | DNI | 23253853 | Cercania2024 |

SERENITY

serenity {  
 encoding = "UTF-8"  
 compress.filenames = true  
 take.screenshots = FOR\_FAILURES *# Cambiar a FOR\_FAILURES para capturar solo en fallos*  
report.include.step.details = false *# Excluir detalles adicionales de pasos*  
report.include.hooks = false *# Excluir tiempos de hooks (@Before y @After)*  
report.show.manual.tests = false *# No mostrar pruebas manuales*  
report.show.released.tests = false *# No mostrar pruebas liberadas*  
report.generate.manual.tests = false *# No generar evidencias de pruebas manuales*  
report.generate.released.tests = false *# No generar evidencias de pruebas liberadas*  
 arn {  
 deviceFarm = "arn:aws:devicefarm:us-west-2:740893539825:testgrid-project:55a2dd70-1fe9-47eb-aa64-e16ffd3371e4"  
 }  
}  
  
 environments {  
 default{  
 login.url = "<https://test.afpintegra.pe/iniciar-sesion>"  
 principal.url = "<https://test.afpintegra.pe/>"  
 outlook.url = "<https://go.microsoft.com/fwlink/p/?LinkID=2125442&amp;clcid=0x409&amp;culture=en-us&amp;country=us>"  
 }  
 dev {  
 login.url = "<https://test.afpintegra.pe/iniciar-sesion>"  
 principal.url = "<https://test.afpintegra.pe/>"  
 outlook.url = "<https://go.microsoft.com/fwlink/p/?LinkID=2125442&amp;clcid=0x409&amp;culture=en-us&amp;country=us>"  
 }  
}

