

## Instrucciones para ejecutar scripts de automatización con Cypress

### Requisitos previos

#### Paso 1: Instalar Node.js

#### Paso 2: Crear tu proyecto

1. Crea una carpeta para tu proyecto
2. Inicializa el proyecto con npm:
  - a. Ejecuta `npm init-y`
  - b. Esto generará un archivo `package.json` en la carpeta del proyecto

#### Paso 3: Instalar Cypress

1. Ejecutar `npm install cypress--save-dev`
2. Abrir Cypress por primera vez con: `npx cypress open`
3. Esto crea una estructura de carpetas en tu proyecto, como `cypress/integration`
4. Verifica la instalación: La ventana de Cypress se abrirá y mostrará proyectos de ejemplo

#### Paso 4: Instalar Cucumber

1. Instala los paquetes necesarios:
  - `npm install--save-dev @badeball/cypress-cucumber-preprocessor cucumber`
2. Configura el preprocesador de Cucumber en Cypress:
  - Abre el archivo de configuración de Cypress, que puede ser `cypress.config.js` o `cypress.config.ts`
  - Crea un directorio para tus archivos `.feature`

#### Paso 5: Configurar GIT y crear un repositorio

1. Inicializar un repositorio Git: En la carpeta del proyecto, ejecuta: `git init`
2. Configura tu repositorio remoto:
  - Crea un repositorio en GitHub.
  - Conecta tu proyecto local al repositorio remoto: `git remote add origin <URL_DE_TU_REPOSITORIO>`
3. Agrega y haz tu primer commit:
  - Agrega los archivos al repositorio: `git add .`
  - Realiza el commit inicial: `git commit -m "Initial commit with Cypress and Cucumber setup"`
4. Sube los cambios:
  - Sube tu proyecto a GitHub: `git push-u origin main`

### Paso 6: Ejecutar Cypress con scripts

1. Abre Cypress con: `npx cypress open`
2. Selecciona tus pruebas desde la interfaz.

## Automatización de casos de prueba con Cypress y Cucumber

### 1. Introducción

Este documento describe los pasos para automatizar los casos de prueba utilizando Cypress con Cucumber en un proyecto desarrollado en Visual Studio Code. Los casos de prueba automatizados en este documento incluyen la validación de diferentes campos y funcionalidades en el formulario de registro y el login de usuario

### 2. Casos de prueba automatizados

A continuación, se detallan los casos de prueba que serán automatizados en este documento:

**Id del caso de prueba:** CP-01

**Nombre del caso de prueba:** Validar que el campo "Full name" acepte mínimo 2 palabras

**Descripción:** Este caso de prueba verifica que el campo "Full name" en el formulario de registro requiera al menos dos palabras (primer nombre y apellido) para habilitar el botón "Sign Up". Se asegura de que el sistema no habilite el botón de registro hasta que el nombre contenga dos palabras

#### Precondiciones:

- El formulario de registro debe estar accesible
- El campo "Full name", "Password" y "Repeat your Password " deben estar habilitados

#### Datos de prueba:

- Full name: Juan
- Email: juan.perez@example.com
- Password: Password123!
- Repeat your password: Password123!

#### Pasos a seguir:

1. Acceder a la URL de la página de registro (<https://test-qa.inlaze.com/auth/sign-up>)
2. En el campo "Full name" ingresar una sola palabra
3. En el campo "Email" ingresar una dirección de correo válida
4. En el campo "Password" ingresar una contraseña válida
5. En el campo "Repeat your password", ingresar nuevamente la contraseña
6. Intenta dar clic en el botón "Sign Up"

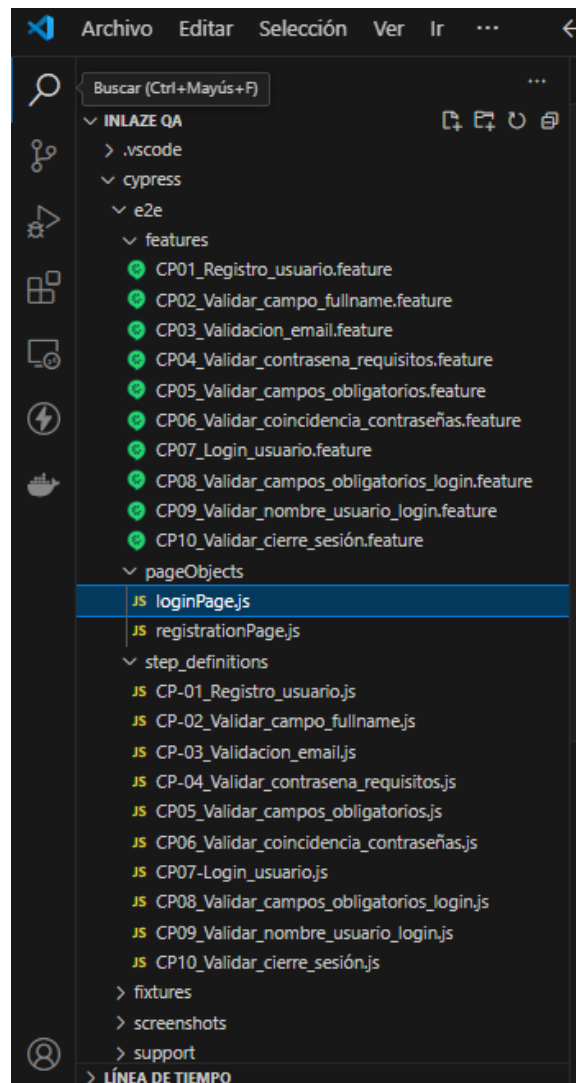
### Resultados esperados:

- El botón “Sign Up” permanece deshabilitado
- El sistema no permite enviar el formulario hasta que se ingrese al menos dos palabras en el campo “Full name” (primer nombre y apellido)
- En el campo “Email” ingresar una dirección de correo válida

### 3. Implementación del caso de prueba

#### Estructura del proyecto

Mi estructura de archivos para este caso queda como sigue:



Script automatizado:

Archivo `.feature:` En el archivo `cypress/e2e/features/CP01_Registro_usuario.feature` se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP01_Registro_usuario.feature
1 Feature: Registro exitoso de usuario con datos válidos
2   Como usuario
3   Quiero registrarme en la plataforma con un nombre, email y contraseñas válidos
4   Para poder acceder a las funcionalidades del sistema correctamente
Open Cypress | Set "@focus"
5 Scenario: Validar que se pueda registrar un usuario con nombre, email y contraseñas válidos
6   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
7   When El usuario ingresa "Juan Perez" en el campo "Full name"
8   And El usuario ingresa "juan.perez@example.com" en el campo "Email"
9   And El usuario ingresa "Password123**" en el campo "Password"
10  And El usuario ingresa "Password123**" en el campo "Repeat your password"
11  And El usuario da clic en el botón "Sign up"
12  Then El sistema debe permitir completar el registro correctamente
13  And El usuario debe ver el mensaje "Successful registration"
14  And El usuario debe ser redirigido a la página de inicio de sesión en "https://test-qa.inlaze.com/auth/sign-in"
```

Archivo de Steps Definitions

En el archivo `cypress/e2e/step_definitions/CP01_Registro_usuario.js` se definieron los pasos del caso de prueba de la siguiente manera:

```
JS CP-01_Registro_usuario.js X
cypress > e2e > step_definitions > JS CP-01_Registro_usuario.js > ...
1 import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2 import RegistrationPage from "../pageObjects/registrationPage"; // Importamos el Page Object
3
4 Given("El usuario navega a la URL del formulario de registro {string}", (url) => {
5   RegistrationPage.visit(url)
6 })
7
8 When("El usuario ingresa {string} en el campo {string}", (inputValue, fieldName) =>{
9   if(fieldName === "Full name"){
10     RegistrationPage.fillFullName(inputValue)
11   } else if(fieldName === "Email"){
12     RegistrationPage.fillEmail(inputValue)
13   } else if(fieldName === "Password"){
14     RegistrationPage.fillPassword(inputValue)
15   } else if(fieldName === "Repeat your password"){
16     RegistrationPage.fillRepeatPassword(inputValue)
17   }
18   cy.wait(300)
19 })
20
21 And("El usuario da clic en el botón {string}", () => {
22   cy.get('[type="submit"]').click()
23 })
24
25 Then("El sistema debe permitir completar el registro correctamente", () => {
26   cy.url().should("include", "/auth/sign-in")
27 })
28
29 And("El usuario debe ver el mensaje {string}", (message) => {
30   cy.get('app-toast > .flex').should("contain.text", message)
31 })
32
33 And("El usuario debe ser redirigido a la página de inicio de sesión en {string}", () => {
34   cy.url().should("include", "/auth/sign-in")
35   .then(()=>{
36     cy.log("¡El usuario ha sido registrado exitosamente!")
37   })
38 })
```

Se realizó el manejo de Page Objects para el registro de usuario como para el Login, quedando los archivos así para el *registrationPage.js*

```
JS registrationPage.js
Cypress > e2e > pageObjects > JS registrationPage.js > ...
1 // pageObjects/registrationPage.js
2
3 class RegistrationPage{
4   visit(){
5     cy.visit("https://test-qa.inlaze.com/auth/sign-up")
6   }
7   fillFullName(fullname){
8     cy.get("#full-name").type(fullname)
9   }
10  fillEmail(email){
11    cy.get("#email").type(email)
12  }
13  fillPassword(password){
14    cy.get("#password").type(password)
15  }
16  fillRepeatPassword(confirmPassword){
17    cy.get("#confirm-password").type(confirmPassword)
18  }
19  checkFormVisibility(){
20    cy.get("form.flex.flex.col.mb-4").should("be.visible")
21  }
22
23  checkFieldEnabled(fieldName){
24    if(fieldName === "Full Name"){
25      cy.get("#full-name").should("be.enabled")
26    } else if(fieldName === "Email"){
27      cy.get("#email").should("be.enabled")
28    } else if(fieldName === "Password"){
29      cy.get("#password").should("be.enabled")
30    } else if(fieldName === "Confirm Password"){
31      cy.get("#confirm-password").should("be.enabled")
32    }
33  }
34
35  checkFieldSelector(fieldName){
36    const selectors = {
37      "Full Name": "#full-name",
38      "Email": "#email",
39      "Password": "#password",
40      "Confirm Password": "#confirm-password"
41    }
42    return selectors[fieldName]
43  }
44
45  clickSubmitButton(){
46    cy.get("[type='submit']").click()
47  }
48
49  checkDuplicateEmailError(){
50    cy.get('app-toast > .flex')
51      .should('be.visible')
52      .then(($toast) => {
53        const message = $toast.text().trim(); // Obtén el texto del mensaje
54        if (message !== "Email inválido") {
55          throw new Error("La prueba falló. El sistema permitió registrar un correo electrónico duplicado.")
56        }
57      })
58  }
59
60  checkButtonState(isEnabled) {
61    const button = cy.get("#btn-primary");
62    button.then($btn => {
63      const isButtonEnabled = !$btn.is(":disabled")
64      if (isButtonEnabled && !isEnabled) {
65        cy.log("ERROR: El botón 'Sign Up' no debería estar habilitado")
66        throw new Error("El botón 'Sign Up' debería estar deshabilitado")
67      } else if (!isButtonEnabled && isEnabled) {
68        // Si el botón está deshabilitado cuando debería estar habilitado
69        cy.log("ERROR: El botón 'Sign Up' debería estar habilitado")
70        throw new Error("El botón 'Sign Up' debería estar habilitado")
71      } else if (!isButtonEnabled && !isEnabled) {
72        // Si el botón está deshabilitado como se esperaba
73        cy.log("El botón 'Sign Up' está deshabilitado como se esperaba")
74      } else {
75        // Si el botón está habilitado como se esperaba
76        cy.log("El botón 'Sign Up' está habilitado como se esperaba")
77      }
78    })
79  }
80
81  checkPasswordMatchError(){
82    // Verificar que el botón "Sign up" esté deshabilitado
83    cy.get("button[type='submit']").should("be.disabled"); // Asegúrate
84
85    // Verificar que el mensaje de error esté visible y contiene el texto esperado
86    cy.get(".label-text-alt")
87      .should("be.visible") // El mensaje de error debe ser visible
88      .then(($toast) => {
89        const message = $toast.text().trim(); // Obtener el texto del mensaje
90        if (message !== "Passwords do not match") {
91          throw new Error("La prueba falló. El sistema permitió registrar contraseñas que no coinciden.")
92        } else {
93          cy.log("No es posible continuar con el registro ya que las contraseñas no coinciden.")
94        }
95      })
96  }
97
98  verifySuccessfulRegistration(){
99    cy.url().should("include", "/auth/sign-in")
100  }
101
102  verifyToastMessage(message){
103    cy.get('app-toast > .flex').should("contain.text", message)
104  }
105 }
106 export default new RegistrationPage;
```

```
JS registrationPage.js
Cypress > e2e > pageObjects > JS registrationPage.js > ...
79
80
81  checkPasswordMatchError(){
82    // Verificar que el botón "Sign up" esté deshabilitado
83    cy.get("button[type='submit']").should("be.disabled"); // Asegúrate
84
85    // Verificar que el mensaje de error esté visible y contiene el texto esperado
86    cy.get(".label-text-alt")
87      .should("be.visible") // El mensaje de error debe ser visible
88      .then(($toast) => {
89        const message = $toast.text().trim(); // Obtener el texto del mensaje
90        if (message !== "Passwords do not match") {
91          throw new Error("La prueba falló. El sistema permitió registrar contraseñas que no coinciden.")
92        } else {
93          cy.log("No es posible continuar con el registro ya que las contraseñas no coinciden.")
94        }
95      })
96  }
97
98  verifySuccessfulRegistration(){
99    cy.url().should("include", "/auth/sign-in")
100  }
101
102  verifyToastMessage(message){
103    cy.get('app-toast > .flex').should("contain.text", message)
104  }
105 }
106 export default new RegistrationPage;
```

Y para el archivo *loginPage.js* queda de la siguiente forma:

```

1 // pageObjects/loginPage.js
2
3 class LoginPage {
4   visit() {
5     cy.visit("https://test-qa.inlaze.com/auth/sign-in")
6   }
7   fillEmail(email) {
8     cy.get("#email").type(email)
9   }
10  fillPassword(password) {
11    cy.get("#password").type(password)
12  }
13  verifyRedirectionToHomepage() {
14    cy.url().should("include", "/panel")
15    .then(() => {
16      cy.log("¡El usuario ha iniciado sesión exitosamente!")
17    })
18  }
19
20  verifyRedirectionToLogin() {
21    cy.url().should("include", "/auth/sign-in")
22    .then(() => {
23      cy.log("¡El usuario ha cerrado sesión exitosamente!")
24    })
25  }
26  clickSignInButton() {
27    cy.get("[type='submit']").click()
28  }
29  clickLogoutButton() {
30    cy.contains("Logout").click()
31  }
32
33  clickProfileImage() {
34    cy.get("img").click()
35  }
36
37  checkUserName() {
38    cy.get(".flex > .font-bold") // Selecciona el elemento que muestra
39    .should("be.visible") // Verifica que sea visible
40    .then(() => {
41      cy.log("Escenario ejecutado exitosamente: El nombre del usuario")
42    })
43  }
44
45  checkErrorMessage(message) {
46    cy.get(".ai-3") // Seleccionamos el campo que contiene el mensaje
47    .should("be.visible") // Verificamos que esté visible
48    .then($toast => {
49      const errorMessage = $toast.text().trim(); // Obtenemos solo el
50      cy.log(errorMessage); // Para depuración: muestra el mensaje de
51      expect(errorMessage).to.include(message); // Verificamos que el
52    });
53  }
54 }
55
56 export default new LoginPage();
57
```

**Id del caso de prueba:** CP-02

**Nombre del caso de prueba:** Validar que el campo nombre acepte mínimo 2 palabras

**Descripción:** Este caso de prueba valida que el campo nombre en el formulario de registro requiera al menos dos palabras (primer nombre y apellido) para habilitar el botón "Sign Up". Se asegura de que el sistema no habilite el botón de registro hasta que el nombre contenga dos palabras

**Precondiciones:**

- El formulario de registro debe estar accesible
- El campo "Full name", "Email", "Password" y "Repeat your Password " deben estar habilitados

**Datos de prueba:**

- Full name: Juan
- Email: juan.perez@example.com
- Password: Password123!
- Repeat your password: Password123!

**Pasos a seguir:**

7. Acceder a la URL de la página de registro (<https://test-qa.inlaze.com/auth/sign-up>)
8. En el campo "Full name", ingresar solo un nombre (Ejemplo: Juan)
9. En el campo "Email", ingresar una dirección de correo válida (Ejemplo: juan.perez@example.com)
10. En el campo "Password" ingresar una contraseña válida que cumpla con los requisitos de longitud y caracteres (Ejemplo: Password123\*)
11. En el campo "Repeat your password", ingresar nuevamente la misma contraseña ingresada en el paso anterior (Ejemplo: Password123\*)
12. Intenta dar clic en el botón "Sign Up"

**Resultados esperados:**

- El botón "Sign Up" debe permanecer deshabilitado
- El sistema no permite enviar el formulario hasta que se ingrese al menos dos palabras en el campo "Full name" (primer nombre y apellido)
- El sistema debe mostrar un mensaje de error indicando que el campo "Full name" debe tener al menos 2 palabras

**Bug relacionado:** BUG-001 (<https://www.notion.so/No-se-muestra-mensaje-de-error-para-Full-name-cuando-no-cumple-con-la-validaci-n-de-m-nimo-dos-pal-16a2455c3d9a805a9d9ce36a34082910?pvs=4>)



Script automatizado:

Archivo `.feature:` En el archivo `cypress/e2e/features/CP02_Validar_campo_fullname.feature` se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP02_Validar_campo_fullname.feature
1 Feature: Verificar que el email cumpla con el formato estandar
  Open Cypress | Set "@focus"
2 Scenario: El botón "Sign up" permanece deshabilitado cuando solo se ingresa una palabra en el campo "Full name"
3   Given El usuario accede a la página de registro "https://test-qa.inlaze.com/auth/sign-up"
4   And El formulario de registro está accesible
5   And Los campos "Full name", "Email", "Password" y "Repeat your password" están habilitados
6   When El usuario ingresa "Juan" en el campo "Full name"
7   And El usuario ingresa "juan.perez@example.com" en el campo "Email"
8   And El usuario ingresa "Password123!" en el campo "Password"
9   And El usuario ingresa "Password123!" en el campo "Repeat your password"
10  Then El botón "Sign up" debe estar deshabilitado
11
12 Open Cypress | Set "@focus"
13 Scenario: El botón "Sign up" se habilita cuando se ingresan dos palabras en el campo "Full name"
14   Given El usuario accede a la página de registro "https://test-qa.inlaze.com/auth/sign-up"
15   And El formulario de registro está accesible
16   And Los campos "Full name", "Email", "Password" y "Repeat your password" están habilitados
17   When El usuario ingresa "Juan Pérez" en el campo "Full name"
18   And El usuario ingresa "juan.perez@example.com" en el campo "Email"
19   And El usuario ingresa "Password123!" en el campo "Password"
20   And El usuario ingresa "Password123!" en el campo "Repeat your password"
21   Then El botón "Sign up" se habilita
```

## Archivo de Steps Definitions

En el archivo `cypress/e2e/step_definitions/CP02_Validar_campo_fullname.js` se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > # CP-02_Validar_campo_fullnamejs > ...
1  import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2  import RegistrationPage from "../pageObjects/registrationPage"; // Importamos el Page Object
3
4  Given("El usuario accede a la página de registro {string}", (url) => {
5    RegistrationPage.visit()
6  })
7
8  And("El formulario de registro está accesible", () => {
9    RegistrationPage.checkFormVisibility()
10 })
11
12 And("Los campos {string}, {string}, {string} y {string} están habilitados", (fullname, email, password, confirmPassword) => {
13   [fullname, email, password, confirmPassword].forEach((field) => {
14     RegistrationPage.checkFieldEnabled(field)
15   })
16 })
17
18 When("El usuario ingresa {string} en el campo {string}", (value, fieldName) =>{
19   const fieldActions = {
20     "Full Name": RegistrationPage.fillFullName,
21     "Email": RegistrationPage.fillEmail,
22     "Password": RegistrationPage.fillPassword,
23     "Repeat your password": RegistrationPage.fillRepeatPassword
24   }
25
26   const action = fieldActions[fieldName]
27   if(action){
28     action(value)
29   }else{
30     throw new Error(`El campo ${fieldName} no está definido`)
31   }
32   cy.wait(300)
33 })
34
35 Then("El botón {string} está deshabilitado", () => {
36   RegistrationPage.checkButtonState(false)
37 })
38
39 Then("El botón {string} se habilita", () => {
40   RegistrationPage.checkButtonState(true)
41 })
```

**Id del caso de prueba:** CP-03

**Nombre del caso de prueba:** Verificar que el email cumpla el formato estándar y sea único en la base de datos

**Descripción:** Este caso de prueba valida que el campo "Email" solo acepte solo emails con el formato estándar de correo electrónico y que el email ingresado no esté ya registrado en la base de datos

**Precondiciones:**

- El formulario de registro debe estar accesible
- El campo "Full name", "Email", "Password" y "Repeat your Password " deben estar habilitados

**Datos de prueba:**

- Full name: Juan Perez
- Email: juan.perez@
- Password: Password123!
- Repeat your password: Password123!

**Pasos a seguir:**

1. Acceder a la URL de la página de registro (<https://test-qa.inlaze.com/auth/sign-up>)
2. En el campo "Full name", ingresar solo un nombre (Ejemplo: Juan)
3. En el campo "Email", ingresar juan.perez@ (email con formato inválido)
4. En el campo "Password" ingresar una contraseña válida que cumpla con los requisitos de longitud y caracteres (Ejemplo: Password123\*)
5. En el campo "Repeat your password", ingresar nuevamente la misma contraseña ingresada en el paso anterior (Ejemplo: Password123\*)
6. Intentar dar clic en el botón "Sign Up"

**Resultados esperados:**

- El sistema debe detectar que el email no cumple con el formato estándar
- El botón "Sign Up" debe permanecer deshabilitado hasta que se ingrese un email con el formato adecuado
- El sistema no debe permitir que se realice el registro de usuario

**Bug relacionado:** BUG-002 (<https://www.notion.so/El-sistema-permite-ingresar-un-email-que-no-cumple-con-el-formato-est-ndar-1692455c3d9a81bebdd8c4adfcc248a7?pvs=4>)

Script automatizado:

Archivo .feature: En el archivo cypress/e2e/features/CP03\_Validacion\_email.feature se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP03_Validacion_email.feature
1 Feature: Verificar que el email cumpla con el formato estándar y sea único en la base de datos
2 Como usuario que desea registrarse
3 Quiero que el sistema valide que el email ingresado
4 Para asegurarme de que el email siga el formato estándar y sea único en la base de datos
  Open Cypress | Set "@focus"
5 Scenario: El sistema detecta un email con formato inválido
6   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
7   When El usuario ingresa "Juan Perez" en el campo "Full name"
8   And El usuario ingresa "juan.perez@" en el campo "Email"
9   And El usuario ingresa "Password123*" en el campo "Password"
10  And El usuario ingresa "Password123*" en el campo "Repeat your password"
11  Then El botón "Sign up" debe estar deshabilitado
  Open Cypress | Set "@focus"
12 Scenario: El sistema valida que el email no esté duplicado en la base de datos
13   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
14   When El usuario ingresa "Juan Perez" en el campo "Full name"
15   And El usuario ingresa "juan.perez@example.com" en el campo "Email"
16   And El usuario ingresa "Password123*" en el campo "Password"
17   And El usuario ingresa "Password123*" en el campo "Repeat your password"
18   And El usuario hace clic en el botón "Sign up"
19   Then El sistema muestra un mensaje de error indicando que el email ya está registrado
```

Archivo de Steps Definitions

En el archivo cypress/e2e/step\_definitions/CP03\_Validacion\_email.js se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP-03_Validacion_email.js > ...
1 import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2 import RegistrationPage from "../pageObjects/registrationPage"; // Importamos el Page Object
3
4 Given("El usuario navega a la URL del formulario de registro {string}", (url) => {
5   RegistrationPage.visit(url)
6 })
7
8 When("El usuario ingresa {string} en el campo {string}", (inputValue, fieldName) => {
9   if(fieldName === "Full name"){
10     RegistrationPage.fillFullName(inputValue)
11   } else if(fieldName === "Email"){
12     RegistrationPage.fillEmail(inputValue)
13   } else if(fieldName === "Password"){
14     RegistrationPage.fillPassword(inputValue)
15   } else if(fieldName === "Repeat your password"){
16     RegistrationPage.fillRepeatPassword(inputValue)
17   }
18   cy.wait(500)
19 })
20
21 Then("El botón {string} debe estar deshabilitado", () => {
22   RegistrationPage.checkButtonState(false)
23 })
24
25 Then("El sistema muestra un mensaje de error indicando que el email ya está registrado", () => {
26   RegistrationPage.checkDuplicateEmailError()
27 })
28
```

**Id del caso de prueba:** CP-04

**Nombre del caso de prueba:** Validar que la contraseña cumpla con los requisitos de longitud y caracteres

**Descripción:** Este caso de prueba valida que la contraseña cumpla con los requisitos de longitud mínima de 8 caracteres y contenga al menos un carácter especial, una mayúscula y un número

**Precondiciones:**

- El formulario de registro debe estar accesible
- El campo "Full name", "Email", "Password" y "Repeat your Password " deben estar habilitados
- El email ya debe existir en la base de datos

**Datos de prueba:**

- Full name: Juan Perez
- Email: [juan.perez@example.com](mailto:juan.perez@example.com)
- Password: Password123!
- Repeat your password: Password123!

**Pasos a seguir:**

1. Acceder a la URL de la página de registro (<https://test-qa.inlaze.com/auth/sign-up>)
2. En el campo "Full name", ingresar Juan Pérez
3. En el campo "Email", ingresar [juan.perez@example.com](mailto:juan.perez@example.com)
4. En el campo "Password" ingresar una contraseña válida que cumpla con los requisitos de longitud y caracteres (Ejemplo: Password123\*)
5. En el campo "Repeat your password", ingresar nuevamente la misma contraseña ingresada en el paso anterior (Ejemplo: Password123\*)
6. Dar clic en el botón "Sign Up"

**Resultados esperados:**

- El botón "Sign Up" está habilitado
- El sistema permite completar el registro con una contraseña válida

**Bug relacionado:** BUG-002 (<https://www.notion.so/El-sistema-permite-registrar-una-contrase-a-que-no-cumple-con-los-requisitos-m-nimos-establecidos-1692455c3d9a80a99171e8b2c40d8df4?pvs=4>)

Script automatizado:

Archivo .feature: En el archivo cypress/e2e/features/CP04\_Validar\_contraseña\_requisitos.feature se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP04_Validar_contraseña_requisitos.feature
D:\Inlaze QA\cypress idar que la contraseña cumpla con los requisitos de longitud y caracteres
2 Como usuario que desea registrarse
3 Quiero que el sistema valide que la contraseña ingresada cumpla con los requisitos de longitud y caracteres
4 Para asegurarme de que cumpla con los requisitos de longitud mínima de 8 caracteres, al menos una letra mayúscula, al menos un número y al menos un carácter especial
5
Open Cypress | Set '@focus'
6
Scenariio: El sistema valida que la contraseña cumple con los requisitos
7
  Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
8
  When El usuario ingresa "Juan Perez" en el campo "Full name"
9
  And El usuario ingresa "juan.perez@example.com" en el campo "Email"
10
  And El usuario ingresa "Password123*" en el campo "Password"
11
  And El usuario ingresa "Password123*" en el campo "Repeat your password"
12
  Then El botón "Sign up" debe estar habilitado
13
Open Cypress | Set '@focus'
14
Scenariio: El sistema rechaza una contraseña sin caracteres especiales
15
  Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
16
  When El usuario ingresa "Juan Perez" en el campo "Full name"
17
  And El usuario ingresa "juan.perez@example.com" en el campo "Email"
18
  And El usuario ingresa "Password123" en el campo "Password"
19
  And El usuario ingresa "Password123" en el campo "Repeat your password"
20
  Then El botón "Sign up" debe estar deshabilitado
21
Open Cypress | Set '@focus'
22
Scenariio: El sistema rechaza una contraseña sin mayúsculas
23
  Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
24
  When El usuario ingresa "Juan Perez" en el campo "Full name"
25
  And El usuario ingresa "juan.perez@example.com" en el campo "Email"
26
  And El usuario ingresa "password123*" en el campo "Password"
27
  And El usuario ingresa "password123*" en el campo "Repeat your password"
28
  Then El botón "Sign up" debe estar deshabilitado
29
Open Cypress | Set '@focus'
30
Scenariio: El sistema rechaza una contraseña sin números
31
  Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
32
  When El usuario ingresa "Juan Perez" en el campo "Full name"
33
  And El usuario ingresa "juan.perez@example.com" en el campo "Email"
34
  And El usuario ingresa "Password*" en el campo "Password"
35
  And El usuario ingresa "Password*" en el campo "Repeat your password"
36
  Then El botón "Sign up" debe estar deshabilitado
```

## Archivo de Steps Definitions

En el archivo `cypress/e2e/step_definitions/CP-04_Validar_contraseña_requisitos.js` se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP-04_Validar_contraseña_requisitos.js > ...
1  import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2  import RegistrationPage from "../pageObjects/registrationPage"; // Importamos el Page Object
3
4  Given("El usuario navega a la URL del formulario de registro {string}", (url) => {
5    | RegistrationPage.visit()
6  })
7
8  When("El usuario ingresa {string} en el campo {string}", (inputValue, fieldName) =>{
9    | if(fieldName === "Full name"){
10     | RegistrationPage.fillFullName(inputValue)
11   } else if(fieldName === "Email"){
12     | RegistrationPage.fillEmail(inputValue)
13   } else if(fieldName === "Password"){
14     | RegistrationPage.fillPassword(inputValue)
15   } else if(fieldName === "Repeat your password"){
16     | RegistrationPage.fillRepeatPassword(inputValue)
17   }
18   | cy.wait(500)
19 })
20
21 Then("El botón {string} debe estar habilitado", () => {
22   | RegistrationPage.checkButtonState(true)
23 })
24
25 Then("El botón {string} debe estar deshabilitado", () => {
26   | RegistrationPage.checkButtonState(false)
27 })
28
29 Then("El botón {string} debe estar deshabilitado", () => {
30   | RegistrationPage.checkButtonState(false)
31 })
32
33 Then("El botón {string} debe estar deshabilitado", () => {
34   | RegistrationPage.checkButtonState(false)
35 })
```

**Id del caso de prueba:** CP-05

**Nombre del caso de prueba:** Comprobar que el formulario no se envíe si los campos obligatorios no están completos

**Descripción:** Este caso de prueba valida que el formulario de registro no se pueda enviar si alguno de los campos obligatorios no está completo

**Precondiciones:**

- El formulario de registro debe estar accesible
- El campo "Full name", "Email", "Password" y "Repeat your Password " deben estar habilitados

**Datos de prueba:**

- Full name: Juan Perez
- Email: dejarlo vacío
- Password: Password123!
- Repeat your password: Password123!

**Pasos a seguir:**

1. Acceder a la URL de la página de registro (<https://test-qa.inlaze.com/auth/sign-up>)
2. En el campo "Full name", ingresar Juan Pérez
3. Dejar vacío el campo email
4. En el campo "Password" ingresar una contraseña válida que cumpla con los requisitos de longitud y caracteres (Ejemplo: Password123\*)
5. En el campo "Repeat your password", ingresar nuevamente la misma contraseña ingresada en el paso anterior (Ejemplo: Password123\*)
6. Intentar dar clic en el botón "Sign Up"

**Resultados esperados:**

- El botón "Sign Up" debe estar deshabilitado



Script automatizado:

Archivo .feature: En el archivo cypress/e2e/features/CP05\_Validar\_campos\_obligatorios.feature se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP05_Validar_campos_obligatorios.feature
1 Feature: Comprobar que el formulario no se envíe si los campos obligatorios no están completos
2 Como usuario que desea registrarse
3 Quiero asegurarme de que el formulario no se pueda enviar si alguno de los campos obligatorios no está completo
4 Para evitar que se registren usuarios con información incompleta
5
6 Open Cypress | Set "@focus"
7 Scenario: El sistema no permite enviar el formulario si el campo "Full name" está vacío
8   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
9   When El usuario deja vacío el campo "Full name"
10  And El usuario ingresa "juan.perez@example.com" en el campo "Email"
11  And El usuario ingresa "Password123*" en el campo "Password"
12  And El usuario ingresa "Password123*" en el campo "Repeat your password"
13  Then El botón "Sign up" debe estar deshabilitado
14
15 Open Cypress | Set "@focus"
16 Scenario: El sistema no permite enviar el formulario si el campo "Email" está vacío
17   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
18   When El usuario ingresa "Juan Perez" en el campo "Full name"
19   And El usuario deja vacío el campo "Email"
20   And El usuario ingresa "Password123*" en el campo "Password"
21   And El usuario ingresa "Password123*" en el campo "Repeat your password"
22   Then El botón "Sign up" debe estar deshabilitado
23
24 Open Cypress | Set "@focus"
25 Scenario: El sistema no permite enviar el formulario si el campo "Password" está vacío
26   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
27   When El usuario ingresa "Juan Perez" en el campo "Full name"
28   And El usuario ingresa "juan.perez@example.com" en el campo "Email"
29   And El usuario deja vacío el campo "Password"
30   And El usuario ingresa "Password123*" en el campo "Repeat your password"
31   Then El botón "Sign up" debe estar deshabilitado
32
33 Open Cypress | Set "@focus"
34 Scenario: El sistema no permite enviar el formulario si el campo "Repeat your password" está vacío
35   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
36   When El usuario ingresa "Juan Perez" en el campo "Full name"
37   And El usuario ingresa "juan.perez@example.com" en el campo "Email"
38   And El usuario ingresa "Password123*" en el campo "Password"
39   And El usuario deja vacío el campo "Repeat your password"
40   Then El botón "Sign up" debe estar deshabilitado
```

## Archivo de Steps Definitions

En el archivo `cypress/e2e/step_definitions/CP05_Validar_campos_obligatorios.js` se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP05_Validar_campos_obligatorios.js > ...
1  import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2  import RegistrationPage from "../pageObjects/registrationPage"; // Importamos el Page Object
3
4  Given("El usuario navega a la URL del formulario de registro {string}", (url) => {
5    RegistrationPage.visit()
6  })
7
8  When("El usuario deja vacio el campo {string}", (action, fieldName) => {
9    if (action === "deja vacio") {
10      if (fieldName === "Full name") {
11        RegistrationPage.fillFullName("")
12      } else if (fieldName === "Email") {
13        RegistrationPage.fillEmail("")
14      } else if (fieldName === "Password") {
15        RegistrationPage.fillPassword("")
16      } else if (fieldName === "Repeat your password") {
17        RegistrationPage.fillRepeatPassword("")
18      }
19    } else if (action === "ingresa") {
20      const inputValue = arguments[2]
21      if (fieldName === "Full name") {
22        RegistrationPage.fillFullName(inputValue)
23      } else if (fieldName === "Email") {
24        RegistrationPage.fillEmail(inputValue)
25      } else if (fieldName === "Password") {
26        RegistrationPage.fillPassword(inputValue)
27      } else if (fieldName === "Repeat your password") {
28        RegistrationPage.fillRepeatPassword(inputValue)
29      }
30    }
31    cy.wait(500);
32  })
33
34  Then("El botón {string} debe estar deshabilitado", () => {
35    RegistrationPage.checkButtonState(false)
36  })
37
```

**Id del caso de prueba:** CP-06

**Nombre del caso de prueba:** Verificar que el sistema informe si las contraseñas ingresadas no coinciden

**Descripción:** Este caso de prueba verifica que el sistema informe al usuario si las contraseñas ingresadas en los campos "Password" y "Repeat your password" no coinciden

**Precondiciones:**

- El formulario de registro debe estar accesible
- El campo "Full name", "Email", "Password" y "Repeat your Password " deben estar habilitados

**Datos de prueba:**

- Full name: Juan Perez
- Email: [juan.perez@example.com](mailto:juan.perez@example.com)
- Password: Password123!
- Repeat your password: Password321\*

**Pasos a seguir:**

1. Acceder a la URL de la página de registro (<https://test-qa.inlaze.com/auth/sign-up>)
2. En el campo "Full name", ingresar Juan Pérez
3. En el campo "Email", ingresar [juan.perez@example.com](mailto:juan.perez@example.com)
4. En el campo "Password" ingresar una contraseña válida que cumpla con los requisitos de longitud y caracteres (Ejemplo: Password123\*)
5. En el campo "Repeat your password", ingresar nuevamente la misma contraseña ingresada en el paso anterior (Ejemplo: Password321\*)
6. Dar clic en el botón "Sign Up"

**Resultados esperados:**

- El botón "Sign Up" permanece deshabilitado
- El sistema muestra el siguiente mensaje: *"Passwords do not match"*

Script automatizado:

Archivo .feature: En el archivo cypress/e2e/features/CP05\_Validar\_campos\_obligatorios.feature se describe el caso de prueba en formato Gherkin:

```
1 Feature: Verificar que el sistema informe si las contraseñas no coinciden
2 Como usuario que desea registrarse
3 Quiero asegurarme de que el sistema informe si las contraseñas ingresadas "Password" y "Repeat your password" no coinciden
4 Para evitar que se registren usuarios con contraseñas diferentes
5
6 Open Cypress | Set '@focus'
7
8 Scenario: El sistema informa que las contraseñas no coinciden
9   Given El usuario navega a la URL del formulario de registro "https://test-qa.inlaze.com/auth/sign-up"
10  When El usuario ingresa "Juan Perez" en el campo "Full name"
11  And El usuario ingresa "juan.perez@example.com" en el campo "Email"
12  And El usuario ingresa "Password123*" en el campo "Password"
13  And El usuario ingresa "Password321*" en el campo "Repeat your password"
14  Then El botón "Sign up" debe estar deshabilitado
15  And El sistema debe mostrar el mensaje "Passwords do not match"
```

## Archivo de Steps Definitions

En el archivo cypress/e2e/step\_definitions/CP06\_Validar\_coincidencia\_contraseñas.js se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP06_Validar_coincidencia_contraseñas.js > ...
1 import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2 import RegistrationPage from "../pageObjects/registrationPage"; // Importamos el Page Object
3
4 Given("El usuario navega a la URL del formulario de registro {string}", (url) => {
5   RegistrationPage.visit()
6 })
7
8 When("El usuario ingresa {string} en el campo {string}", (inputValue, fieldName) => {
9   if(fieldName === "Full name"){
10     RegistrationPage.fillFullName(inputValue)
11   } else if(fieldName === "Email"){
12     RegistrationPage.fillEmail(inputValue)
13   } else if(fieldName === "Password"){
14     RegistrationPage.fillPassword(inputValue)
15   } else if(fieldName === "Repeat your password"){
16     RegistrationPage.fillRepeatPassword(inputValue)
17   }
18   cy.wait(500)
19 })
20
21 Then("El botón {string} debe estar deshabilitado", () => {
22   RegistrationPage.checkButtonState(false)
23 })
24
25 And("El sistema debe mostrar el mensaje {string}", () => {
26   RegistrationPage.checkPasswordMatchError()
27 })
```

## Login de Usuario

**Id del caso de prueba:** CP-07

**Nombre del caso de prueba:** Verificar que el usuario pueda loguearse con el email y contraseñas correctos

**Descripción:** Este caso valida que el sistema permita al usuario iniciar sesión correctamente cuando se ingresan credenciales válidas

### Precondiciones:

- El formulario de login debe estar accesible
- El usuario debe estar registrado en la plataforma con las credenciales válidas

### Datos de prueba:

- Email: `juan.perez@example.com`
- Password: `Password123*`

### Pasos a seguir:

1. Acceder a la URL de la página de login `https://test-qa.inlaze.com/auth/login`
2. En el campo email, ingresar `juan.perez@example.com`
3. En el campo password ingresar `Password123*`
4. Hacer clic en el botón "Sign In"

### Resultado esperado:

- El sistema valida las credenciales ingresadas
- El usuario es redirigido a la página principal de la plataforma

Script automatizado:

Archivo .feature: En el archivo cypress/e2e/features/ CP07\_Login\_usuario.feature se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP07_Login_usuario.feature
1 Feature: Verificar que el usuario pueda loguearse con el email y contraseña correctos
2 Como usuario registrado
3 Quiero poder loguearme con mi email y contraseña
4 Para poder acceder a la plataforma
5
6 # Escenario para login exitoso con credenciales correctas
7 Open Cypress | Set '@focus'
8 Scenario: Ingresar con email y contraseña correctos
9   Given El usuario navega a la URL del login "https://test-qa.inlaze.com/auth/sign-in"
10  When El usuario ingresa "andres.perez@example.com" en el campo "Email"
11  And El usuario ingresa "Password123*" en el campo "Password"
12  And El usuario hace clic en el botón "Sign In"
13  Then El usuario es redirigido a la página principal de la plataforma "https://test-qa.inlaze.com/panel"
14
15 # Escenario para contraseña incorrecta
16 Open Cypress | Set '@focus'
17 Scenario: Verificar que el sistema no permita iniciar sesión con la contraseña incorrecta
18   Given El usuario navega a la URL del login "https://test-qa.inlaze.com/auth/sign-in"
19   When El usuario ingresa "andres.perez@example.com" en el campo "Email"
20   And El usuario ingresa "Password1234*" en el campo "Password"
21   And El usuario hace clic en el botón "Sign In"
22   Then El sistema muestra el mensaje "Password doesn't match"
23
24 # Escenario para usuario no registrado
25 Open Cypress | Set '@focus'
26 Scenario: Verificar que el sistema no permita iniciar sesión con un usuario no registrado
27   Given El usuario navega a la URL del login "https://test-qa.inlaze.com/auth/sign-in"
28   When El usuario ingresa "inc.perez@example.com" en el campo "Email"
29   And El usuario ingresa "Password123*" en el campo "Password"
30   And El usuario hace clic en el botón "Sign In"
31   Then El sistema muestra el siguiente mensaje "User not found"
```

Archivo de Steps Definitions

En el archivo cypress/e2e/step\_definitions/CP07\_Login\_usuario.js se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP07_Login_usuario.js > ...
1 import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2 import LoginPage from "../pageObjects/loginPage"; // Importamos el Page Object
3
4 Given("El usuario navega a la URL del login {string}", (url) => {
5   LoginPage.visit()
6 })
7
8 When("El usuario ingresa {string} en el campo {string}", (inputValue, fieldName) => {
9   if(fieldName === "Email"){
10     LoginPage.fillEmail(inputValue)
11   } else if(fieldName === "Password"){
12     LoginPage.fillPassword(inputValue)
13   }
14   cy.wait(1000)
15 })
16
17 And("El usuario hace clic en el botón {string}", () => {
18   LoginPage.clickSignInButton()
19 })
20
21 Then("El usuario es redirigido a la página principal de la plataforma {string}", () => {
22   LoginPage.verifyRedirectionToHomepage()
23 })
24
25 Then("El sistema muestra el mensaje {string}", () => {
26   LoginPage.checkErrorMessage("Password doesn't match")
27 })
28
29 Then("El sistema muestra el siguiente mensaje {string}", () => {
30   LoginPage.checkErrorMessage("User not found")
31 })
```

**Id del caso de prueba:** CP-08

**Nombre del caso de prueba:** Validar que el formulario de login no se envíe si los campos no están completos

**Descripción:** Este caso valida que el sistema no permita enviar el formulario si alguno de los campos requeridos está vacío

**Precondiciones:**

- El usuario debe estar registrado
- El formulario del login debe estar accesible

**Datos de prueba:**

- Email: `juan.perez@example.com`
- Password: `Password123*`

**Pasos a seguir:**

1. Acceder a la URL de la página de login `https://test-qa.inlaze.com/auth/sign-in`
2. En el campo email ingresar `pedro.alonso@example.com`
3. Dejar el campo "Password" vacío
4. Intentar dar clic en el botón "Sign In"

**Resultado esperado:**

- El botón Sign In debe permanecer deshabilitado mientras los campos no estén completos
- El sistema debe mostrar un mensaje indicando que el campo Password es obligatorio

Script automatizado:

Archivo .feature: En el archivo cypress/e2e/features/CP08\_Validar\_campos\_obligatorios\_login.feature se describe el caso de prueba en formato Gherkin:

```
registrationPage.js • JS loginPage.js • CP08_Validar_campos_obligatorios_login.feature X
cypress > e2e > features > CP08_Validar_campos_obligatorios_login.feature
1 Feature: Validar que el formulario del login no se envíe si los campos no están completos
2 Como usuario que desea ingresar
3 Quiero asegurarme de que el formulario del login no se envíe si los campos no están completos
4 Para verificar que el sistema me solicita completar los campos obligatorios
5
6 Open Cypress | Set "@focus"
7 Scenario: El formulario no se envía cuando el campo "Password" está vacío
8   Given El usuario navega a la URL del login "https://test-qa.inlaze.com/auth/sign-in"
9   When El usuario ingresa "andres.perez@example.com" en el campo "Email"
10  And El usuario deja vacío el campo "Password"
11  Then El botón "Sign in" debe estar deshabilitado
12
13 Open Cypress | Set "@focus"
14 Scenario: El formulario no se envía cuando el campo "Email" está vacío
15   Given El usuario navega a la URL del login "https://test-qa.inlaze.com/auth/sign-in"
16   When El usuario deja vacío el campo "Email"
17   And El usuario ingresa "Password123*" en el campo "Password"
18   Then El botón "Sign in" debe estar deshabilitado
```

## Archivo de Steps Definitions

En el archivo cypress/e2e/step\_definitions/CP08\_Validar\_campos\_obligatorios\_login.js se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP08_Validar_campos_obligatorios_login.js > ...
1 import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2 import LoginPage from "../pageObjects/loginPage"; // Importamos el Page Object
3
4 Given("El usuario navega a la URL del login {string}", (url) => {
5   LoginPage.visit()
6 })
7
8 When("El usuario deja vacío el campo {string}", (action, fieldName) => {
9   if (action === "deja vacío") {
10     if (fieldName === "Email") {
11       LoginPage.fillEmail("")
12     } else if (fieldName === "Password") {
13       LoginPage.fillPassword("")
14     }
15   } else if (action === "ingresa") {
16     const inputValue = arguments[2]
17     if (fieldName === "Email") {
18       LoginPage.fillEmail(inputValue)
19     } else if (fieldName === "Password") {
20       LoginPage.fillPassword(inputValue)
21     }
22   }
23   cy.wait(1000)
24 })
```



**Id del caso de prueba:** CP-09

**Nombre del caso de prueba:** Comprobar que al ingresar se muestre el nombre del usuario

**Descripción:** Este caso valida que al iniciar sesión exitosamente, el sistema confirme que la sesión pertenece al usuario logueado

**Precondiciones:**

- El usuario debe estar registrado
- El formulario del login debe estar accesible

**Datos de prueba:**

- Email: `juan.lorenzo @example.com`
- Password: `Password123*`

**Pasos a seguir:**

1. Acceder a la URL de la página de login `https://test-qa.inlaze.com/auth/sign-in`
2. En el campo email ingresar `juan.lorenzo@example.com`
3. En el campo Password, ingresar `Password123*`
4. Hacer clic en el botón "Sign In"

**Resultado esperado:**

- El sistema valida las credenciales ingresadas
- El usuario es redirigido a la página principal

Script automatizado:

Archivo `.feature:` En el archivo `cypress/e2e/features/CP09_Validar_nombre_usuario_login.feature` se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP09_Validar_nombre_usuario_login.feature
1 Feature: Comprobar que al ingresar se muestre el nombre de usuario
2 Como usuario que desea ingresar
3 Quiero que el sistema muestre mi nombre al iniciar sesión exitosamente
4 Para confirmar que la sesión pertenece a mi usuario logueado
5
Open Cypress | Set "@focus"
6 Scenario: Validar que el sistema muestre el nombre de usuario después de iniciar sesión
7   Given El usuario navega a la URL del login "https://test-qa.inlaze.com/auth/sign-in"
8   When El usuario ingresa "andres.perez@example.com" en el campo "Email"
9   And El usuario ingresa "Password123*" en el campo "Password"
10  And El usuario hace clic en el botón "Sign In"
11  Then El usuario es redirigido a la página principal de la plataforma "https://test-qa.inlaze.com/panel"
12  And El sistema muestra el nombre de usuario "Andres Perez"
```

## Archivo de Steps Definitions

En el archivo `cypress/e2e/step_definitions/CP09_Validar_nombre_usuario_login.js` se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP09_Validar_nombre_usuario_login.js > ...
1 import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2 import LoginPage from "../pageObjects/loginPage"; // Importamos el Page Object
3
4 Given("El usuario navega a la URL del login {string}", (url) => {
5   LoginPage.visit()
6 })
7
8 When("El usuario ingresa {string} en el campo {string}", (inputValue, fieldName) =>{
9   if(fieldName === "Email"){
10     LoginPage.fillEmail(inputValue)
11   } else if(fieldName === "Password"){
12     LoginPage.fillPassword(inputValue)
13   }
14   cy.wait(1000)
15 })
16
17 And("El usuario hace clic en el botón {string}", () => {
18   LoginPage.clickSignInButton()
19   cy.wait(1000)
20 })
21
22 Then("El usuario es redirigido a la página principal de la plataforma {string}", () => {
23   LoginPage.verifyRedirectionToHomepage()
24   cy.wait(1000)
25 })
26
27 And("El sistema muestra el nombre de usuario {string}", (userName) => {
28   LoginPage.checkUserName(userName)
29   cy.wait(1000)
30 })
```

**Id del caso de prueba:** CP-10

**Nombre del caso de prueba:** Verificar que la plataforma permita cerrar la sesión correctamente

**Descripción:** Este caso valida que el usuario pueda cerrar sesión correctamente

**Precondiciones:**

- El usuario debe haber iniciado sesión en la plataforma.

**Datos de prueba:**

- Email: `juan.lorenzo@example.com`
- Password: `Password123*`

**Pasos a seguir:**

5. Acceder a la URL de la página de login <https://test-qa.inlaze.com/auth/sign-in> con las credenciales válidas
  - Email: [juan.lorenzo@example.com](mailto:juan.lorenzo@example.com)
  - Password: `Password123*`
6. Hacer clic en la imagen de perfil que se encuentra en la parte superior derecha
7. Hacer clic en "Logout"

**Resultado esperado:**

- El sistema cierra la sesión actual
- El usuario es redirigido a la página del login

Script automatizado:

Archivo `.feature`: En el archivo `cypress/e2e/features/CP10_Validar_cierre_sesión.feature` se describe el caso de prueba en formato Gherkin:

```
cypress > e2e > features > CP10_Validar_cierre_sesión.feature
1 Feature: Verificar que la plataforma permita cerrar la sesión correctamente
2 Como usuario que desea cerrar sesión
3 Quiero que el sistema permita cerrar la sesión de manera correcta
4 Para confirmar que la sesión ha sido cerrada exitosamente
5
Open Cypress | Set "@focus"
6 Scenario: El usuario cierra sesión correctamente
7   Given El usuario navega a la URL del login "https://test-qa.inlaze.com/auth/sign-in"
8   When El usuario ingresa "andres.perez@example.com" en el campo "Email"
9   And El usuario ingresa "Password123*" en el campo "Password"
10  And El usuario hace clic en el botón "Sign In"
11  And El usuario hace clic en la imagen de perfil
12  And El usuario hace clic en la opción "Logout"
13  Then El usuario es redirigido a la página de login "https://test-qa.inlaze.com/auth/sign-in"
```

### Archivo de Steps Definitions

En el archivo `cypress/e2e/step_definitions/CP10_Validar_cierre_sesión.js` se definieron los pasos del caso de prueba de la siguiente manera:

```
cypress > e2e > step_definitions > JS CP10_Validar_cierre_sesión.js > ...
1 import {Given, When, Then, And} from "cypress-cucumber-preprocessor/steps"
2 import LoginPage from "../pageObjects/loginPage"; // Importamos el Page Object
3
4 Given("El usuario navega a la URL del login {string}", (url) => {
5   LoginPage.visit()
6 })
7
8 When("El usuario ingresa {string} en el campo {string}", (inputValue, fieldName) =>{
9   if(fieldName === "Email"){
10     LoginPage.fillEmail(inputValue)
11   } else if(fieldName === "Password"){
12     LoginPage.fillPassword(inputValue)
13   }
14   cy.wait(1000)
15 })
16
17 And("El usuario hace clic en el botón {string}", () => {
18   LoginPage.clickSignInButton()
19   cy.wait(1000)
20 })
21
22 And("El usuario hace clic en la imagen de perfil", () => {
23   LoginPage.clickProfileImage()
24   cy.wait(600)
25 })
26
27 And("El usuario hace clic en la opción {string}", () => {
28   LoginPage.clickLogoutButton()
29   cy.wait(1000)
30 })
31
32 Then("El usuario es redirigido a la página de login {string}", () => {
33   LoginPage.verifyRedirectionToLogin()
34 })
```