

Proyecto: Ingeniería de Requisitos aplicada a la
plataforma web “Home Passion”
Documento de Requisitos del Sistema

Versión V1.0.0

Fecha: 26/07/024

Realizado por: Arias Sebastián, Carrera Nahir, Pullaguari
Axel (GRUPO 1)
Tutor: Ruiz, Jenny

Link: <https://youtu.be/2un5LaGhnYo>

Contenido

Introducción.....	2
Objetivo General	2
Objetivos Específicos	2
Desarrollo.....	3
Conclusión.....	13
Recomendaciones	13

Introducción

En el desarrollo de aplicaciones web modernas, la autenticación de usuarios es un componente crítico que garantiza la seguridad y personalización del contenido. Para asegurar la fiabilidad y eficiencia del sistema de autenticación, es esencial realizar pruebas exhaustivas. En este informe, se detalla el proceso y resultados de las pruebas del sistema de login desarrollado con React y Next.js, utilizando las herramientas Cucumber y Selenium para la automatización de pruebas.

Objetivo General

Validar nuestro requisito funcional del DRS , en este caso el Login en donde buscamos evaluar la funcionalidad y robustez del sistema de autenticación de usuarios implementado en una aplicación web construida con React y Next.js, mediante la utilización de pruebas automatizadas con Cucumber y Selenium.

Objetivos Específicos

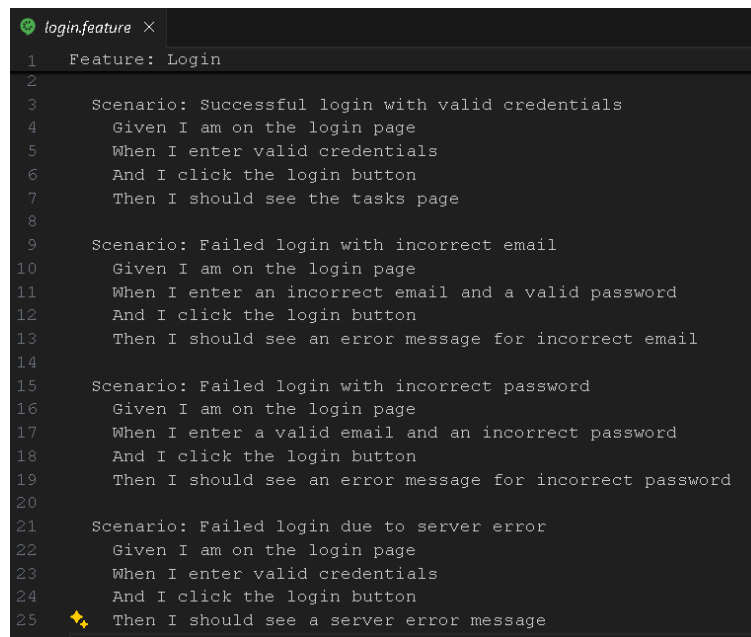
- 1. Verificar la correcta funcionalidad del formulario de login:**
 - Asegurar que los usuarios pueden ingresar sus credenciales y acceder al sistema sin errores.
 - Validar que se manejen adecuadamente los errores de autenticación, tales como contraseñas incorrectas o usuarios inexistentes.
- 2. Garantizar la compatibilidad del sistema en diferentes navegadores y dispositivos:**
 - Ejecutar las pruebas en múltiples navegadores (Chrome, Firefox, Edge) y dispositivos (escritorios, tablets, móviles) para asegurar una experiencia de usuario consistente.
- 3. Evaluar la seguridad del proceso de autenticación:**

- Comprobar que las contraseñas y otros datos sensibles se manejan de manera segura.
- Verificar que las sesiones de usuario se gestionan correctamente y que se implementan medidas adecuadas contra ataques comunes como el Cross-Site Scripting (XSS) y el Cross-Site Request Forgery (CSRF).

Desarrollo

Tras la simulación de la fase de desarrollo para poder crear escenarios para evaluar mediante la ayuda de cucumber y selenium, se ha logrado una buena implementación dentro del entorno de desarrollo, la cohesión entre los scripts para generar los reportes, así como la correcta interpretación de los mismos.

Vamos a abordar los resultados a continuación. Como primera parte tenemos la creación de los escenarios, los cuales nos ayudan mediante el lenguaje natural para un mejor entendimiento de lo que esperamos como resultado de los test.



```
login.feature X
1  Feature: Login
2
3      Scenario: Successful login with valid credentials
4          Given I am on the login page
5          When I enter valid credentials
6          And I click the login button
7          Then I should see the tasks page
8
9      Scenario: Failed login with incorrect email
10         Given I am on the login page
11         When I enter an incorrect email and a valid password
12         And I click the login button
13         Then I should see an error message for incorrect email
14
15     Scenario: Failed login with incorrect password
16         Given I am on the login page
17         When I enter a valid email and an incorrect password
18         And I click the login button
19         Then I should see an error message for incorrect password
20
21     Scenario: Failed login due to server error
22         Given I am on the login page
23         When I enter valid credentials
24         And I click the login button
25         ✨ Then I should see a server error message
26
```

Ilustración 1 Texto en formato Gherkin de los escenarios.

```

login.steps.js X
1  const { Given, When, Then, Before, After } = require('@cucumber/cucumber');
2  const { Builder, By, until } = require('selenium-webdriver');
3  const fs = require('fs');
4  const path = require('path');
5  const assert = require('assert');
6
7  let driver;
8  const screenshotsDir = path.resolve(__dirname, 'screenshots');
9  if (!fs.existsSync(screenshotsDir)) {
10     fs.mkdirSync(screenshotsDir);
11 }
12
13 Before(async function () {
14     driver = await new Builder().forBrowser('chrome').build();
15 });
16
17 After(async function (scenario) {
18     const screenshotPath = path.join(screenshotsDir, `screenshot-${Date.now()}.png`);
19     const screenshot = await driver.takeScreenshot();
20     fs.writeFileSync(screenshotPath, screenshot, 'base64');
21     console.log(`Captura de pantalla guardada en ${screenshotPath}`);
22     await driver.quit();
23 });
24
25 Given('I am on the login page', async function () {
26     await driver.get('http://localhost:3000/login');
27     const screenshotPath = path.join(screenshotsDir, `screenshot-${Date.now()}.png`);
28     const screenshot = await driver.takeScreenshot();
29     fs.writeFileSync(screenshotPath, screenshot, 'base64');
30     console.log(`Captura de pantalla guardada en ${screenshotPath}`);
31 });
32
33 When('I enter valid credentials', async function () {
34     const emailField = await driver.findElement(By.name('email'));
35     const passwordField = await driver.findElement(By.name('password'));
36     await emailField.sendKeys('user@example.com');
37     await passwordField.sendKeys('password');
38     const screenshotPath = path.join(screenshotsDir, `screenshot-${Date.now()}.png`);
39     const screenshot = await driver.takeScreenshot();
40     fs.writeFileSync(screenshotPath, screenshot, 'base64');
41     console.log(`Captura de pantalla guardada en ${screenshotPath}`);
42 });

```

Ilustración 2 Preguntas en código

Por motivos de compatibilidad se realizó la creación de los escenarios en el idioma inglés. A continuación la implementación del script se lo genera bajo los 4 escenarios creados. Escenarios:

1. Ingreso satisfactorio con credenciales validas
2. Ingreso fallido proporcionando un correo no valido
3. Ingreso fallido proporcionando una contraseña incorrecta
4. Escenario donde la pagina web no carga por problemas del servidor

Para una correcta implementación del script se debe tomar muy en cuenta lo que buscamos obtener como resultados, la herramienta de cucumber dentro de nuestro IDE a sido de gran ayuda, obteniendo los siguientes resultados a manera de las capturas, cabe destacar que el archivo obtenido al final en formato json nos permite mediante otro script para generar los reportes tener una visión mas amena y entendible.



Ilustración 3 Captura de la parte inicial del informe.

Como se puede observar a primera vista el reporte que genera nos menciona los escenarios puestos a prueba y el porcentaje que cada tipo de resultado arroja la aplicación. Como se ve en el ejemplo tenemos 3 tipos de resultados “Passed Scenarios”, “Failed Scenarios”, “Undefined Scenarios”. En el mismo reporte tenemos el desglose de los escenarios y sus respectivos pasos. Para el ejemplo del uso de la herramienta se hizo a propósito que pase un escenario, que dos fallen y uno sea encuentre indefinido, y demostrar el potencial de la aplicación.

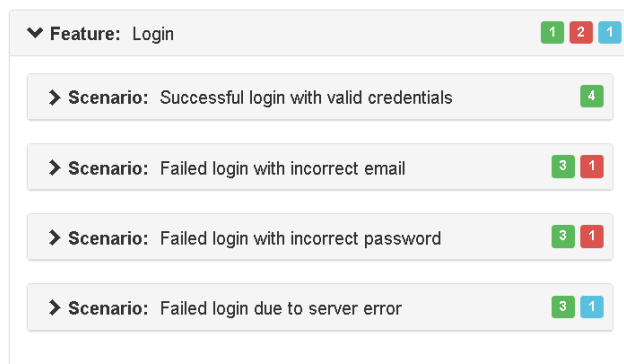


Ilustración 4 Captura de la parte del resumen de los escenarios.

Primer Escenario:

▼ **Scenario:** Successful login with valid credentials

4

✓ Given I am on the login page	1s 481ms
✓ When I enter valid credentials	859ms
✓ And I click the login button	131ms
✓ Then I should see the tasks page	442ms

Ilustración 5 Captura de la parte del resumen de los escenarios.

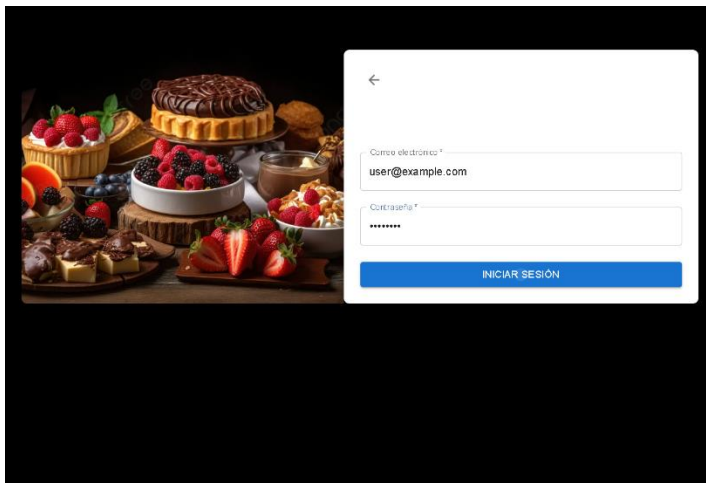


Ilustración 6 Captura de la parte del resumen de los escenarios.

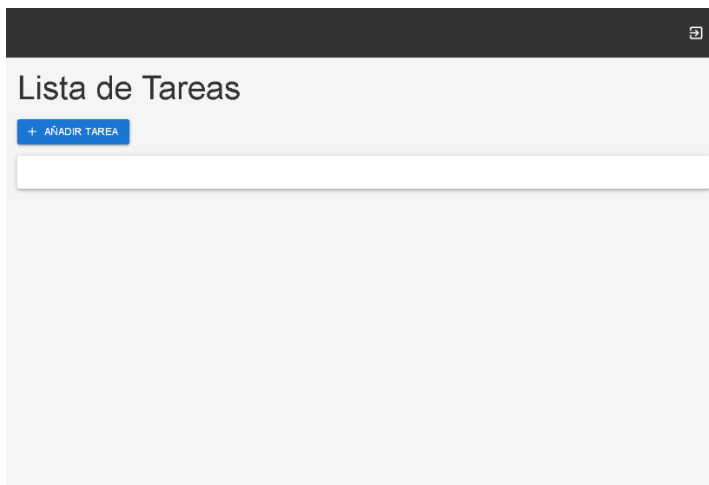


Ilustración 7 Captura de la parte del resumen de los escenarios.

Segundo Escenario:

Scenario: Failed login with incorrect email 3 1

✓ Given I am on the login page	1s 430ms
✓ When I enter an incorrect email and a valid password	953ms
✓ And I click the login button	117ms
✗ Then I should see an error message for incorrect email Show Error -	523ms

```
NoSuchElementException: no such element: Unable to locate element  
(Session info: chrome=127.0.6533.72)  
    at Object.throwDecodedError (D:\ESPE\5. Quinto Semestre\  
    at parseHttpResponse (D:\ESPE\5. Quinto Semestre\INGENIE  
    at Executor.execute (D:\ESPE\5. Quinto Semestre\INGENIER  
    at process.processTicksAndRejections (node:internal/proc  
    at async Driver.execute (D:\ESPE\5. Quinto Semestre\INGE  
    at async toWireValue (D:\ESPE\5. Quinto Semestre\INGENIE  
    at async D:\ESPE\5. Quinto Semestre\INGENIERIA DE REQUIS  
    at async forEachKey (D:\ESPE\5. Quinto Semestre\INGENIER  
    at async convertKeys (D:\ESPE\5. Quinto Semestre\INGENIE  
    at async Driver.execute (D:\ESPE\5. Quinto Semestre\INGE
```

Ilustración 8 Captura de la parte del resumen de los escenarios.

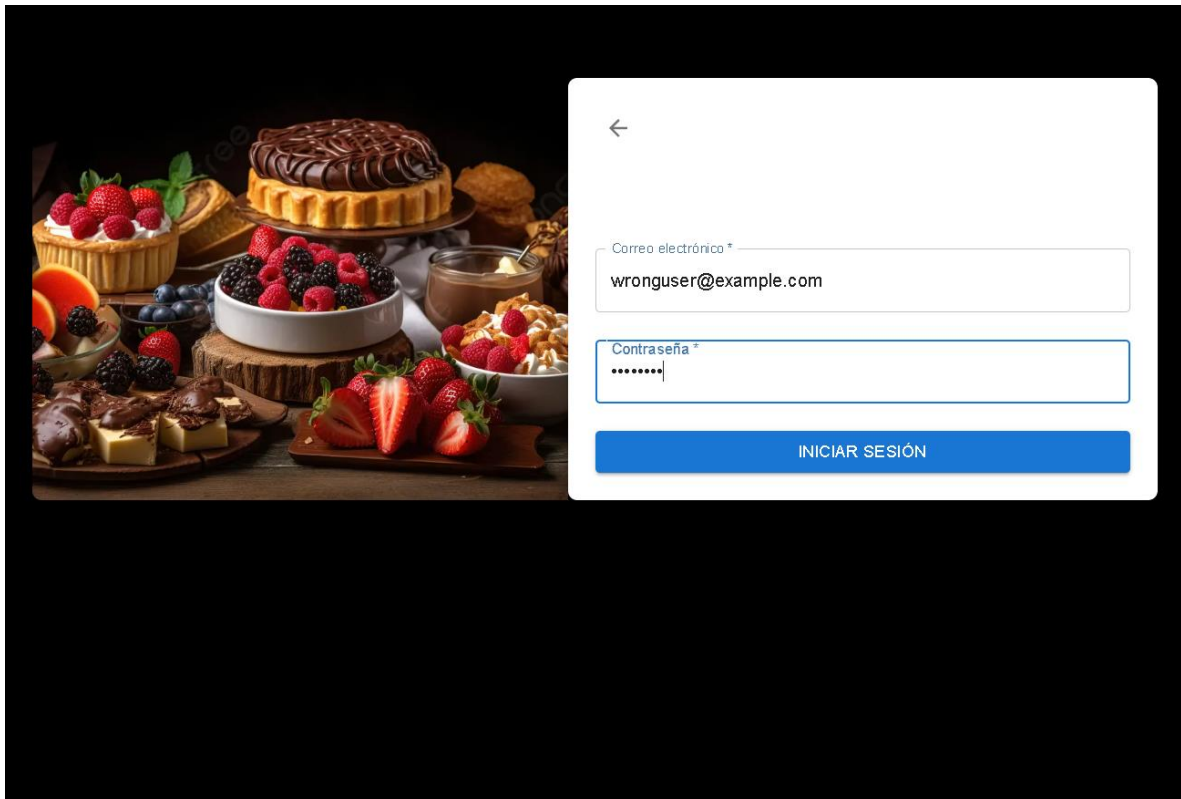


Ilustración 9 Captura de la parte del resumen de los escenarios.

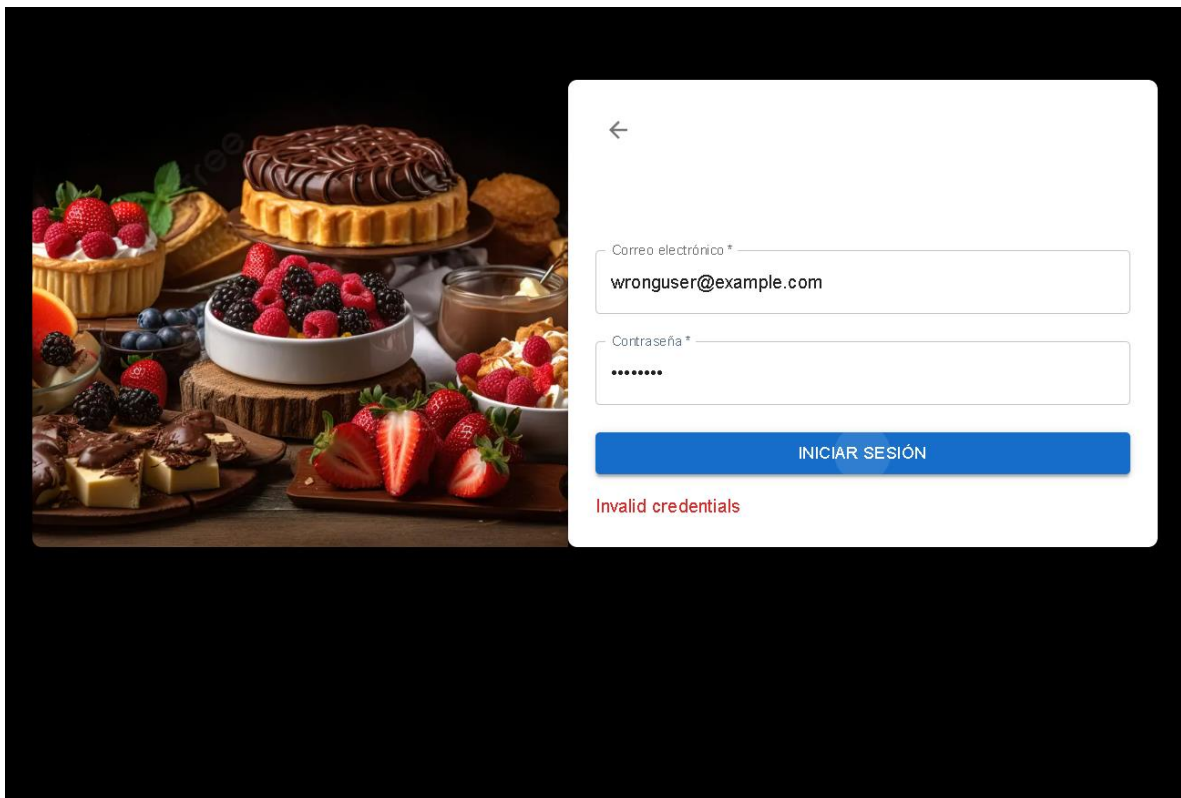


Ilustración 10 Captura de la parte del resumen de los escenarios.

Tercer Escenario:

Scenario: Failed login with incorrect password 3 1

✓ Given I am on the login page	176ms
✓ When I enter a valid email and an incorrect password	2s 260ms
✓ And I click the login button	115ms
✗ Then I should see an error message for incorrect password Show Error -	475ms

```
NoSuchElementException: no such element: Unable to locate element (Session info: chrome=127.0.6533.72)
  at Object.throwDecodedError (D:\ESPE\5. Quinto Semestre\
  at parseHttpResponse (D:\ESPE\5. Quinto Semestre\INGENIE
  at Executor.execute (D:\ESPE\5. Quinto Semestre\INGENIER
  at process.processTicksAndRejections (node:internal/proc
  at async Driver.execute (D:\ESPE\5. Quinto Semestre\INGE
  at async toWireValue (D:\ESPE\5. Quinto Semestre\INGENIE
  at async D:\ESPE\5. Quinto Semestre\INGENIERIA DE REQUIS
  at async forEachKey (D:\ESPE\5. Quinto Semestre\INGENIER
  at async convertKeys (D:\ESPE\5. Quinto Semestre\INGENIE
  at async Driver.execute (D:\ESPE\5. Quinto Semestre\INGE
```

Ilustración 11 Captura de la parte del resumen de los escenarios.

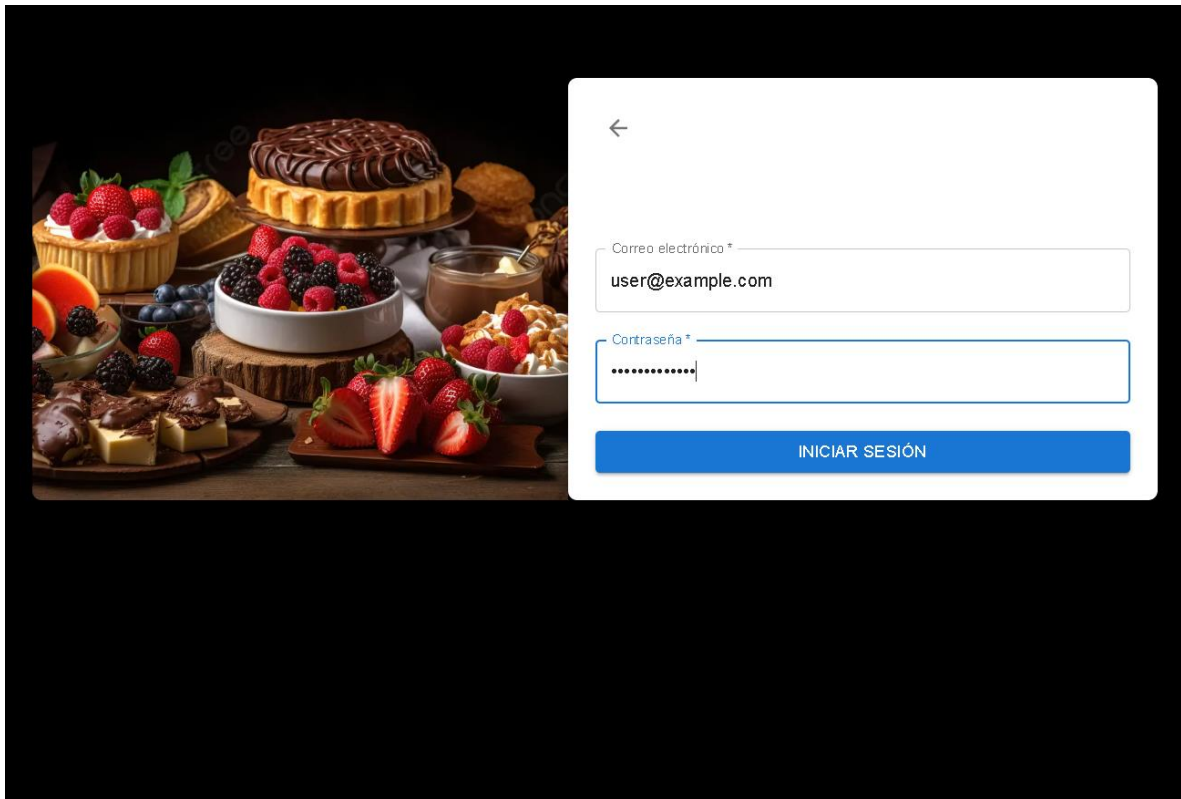


Ilustración 12 Captura de la parte del resumen de los escenarios.

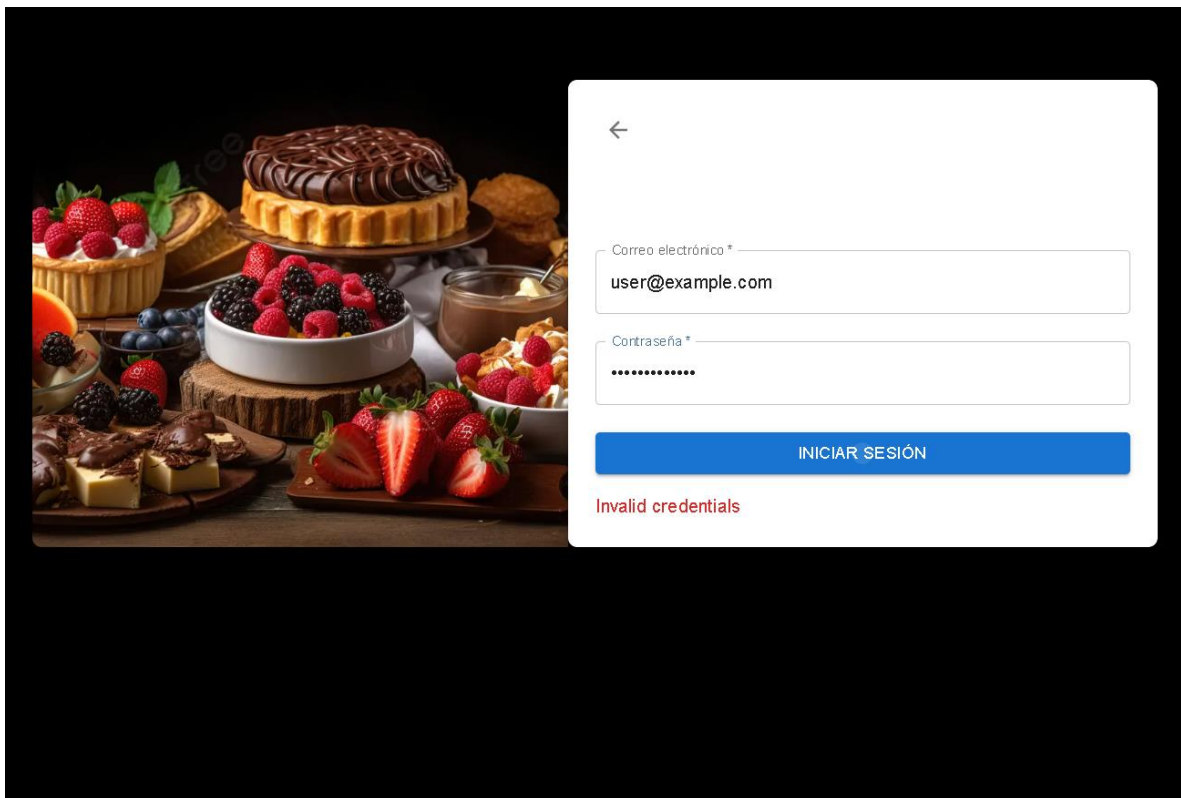


Ilustración 13 Captura de la parte del resumen de los escenarios.

Cuarto Escenario:

▼ Scenario: Failed login due to server error 3 1

✓ Given I am on the login page 1s 661ms

✓ When I enter valid credentials 1s 48ms

✓ And I click the login button 153ms

⚡ Then I should see a server error message < 1ms

```
Then(/^I should see a server error message$/, function(callback) {
  // Write code here that turns the phrase above into concrete steps
  callback(null, 'pending');
});
```

Ilustración 14 Captura de la parte del resumen de los escenarios.

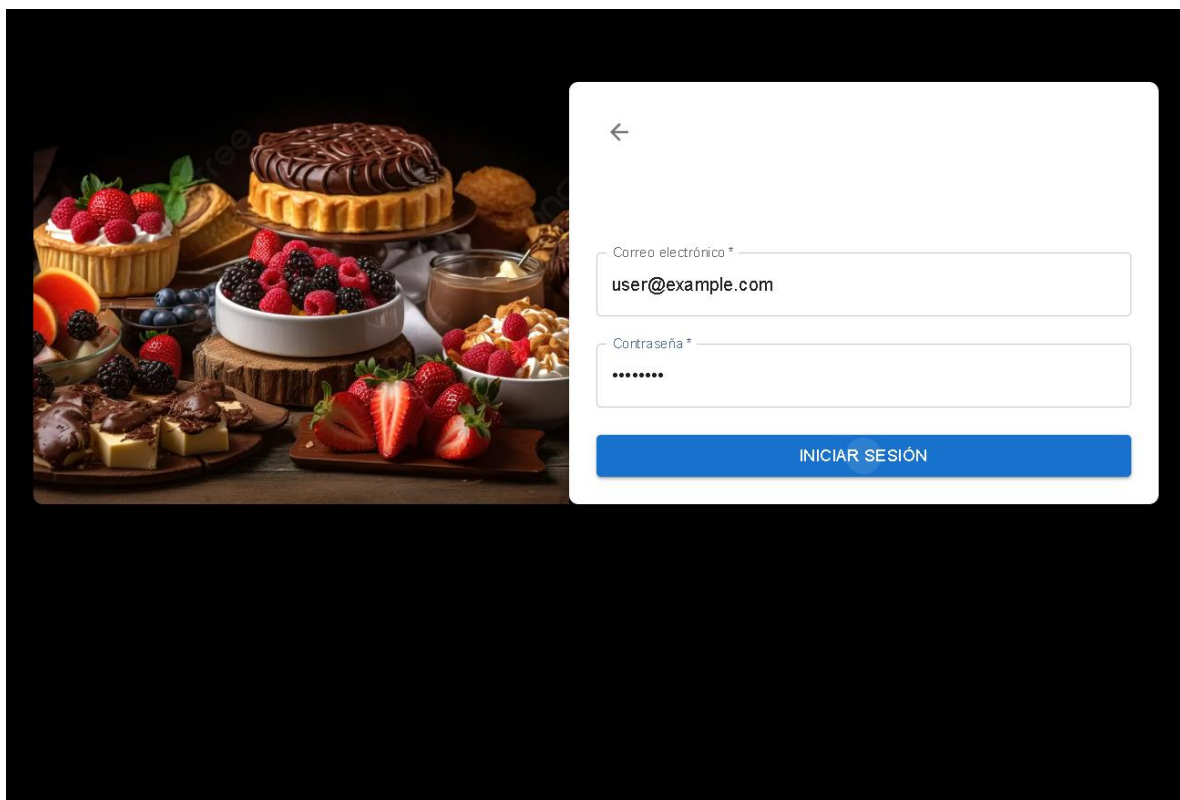


Ilustración 15 Captura de la parte del resumen de los escenarios.

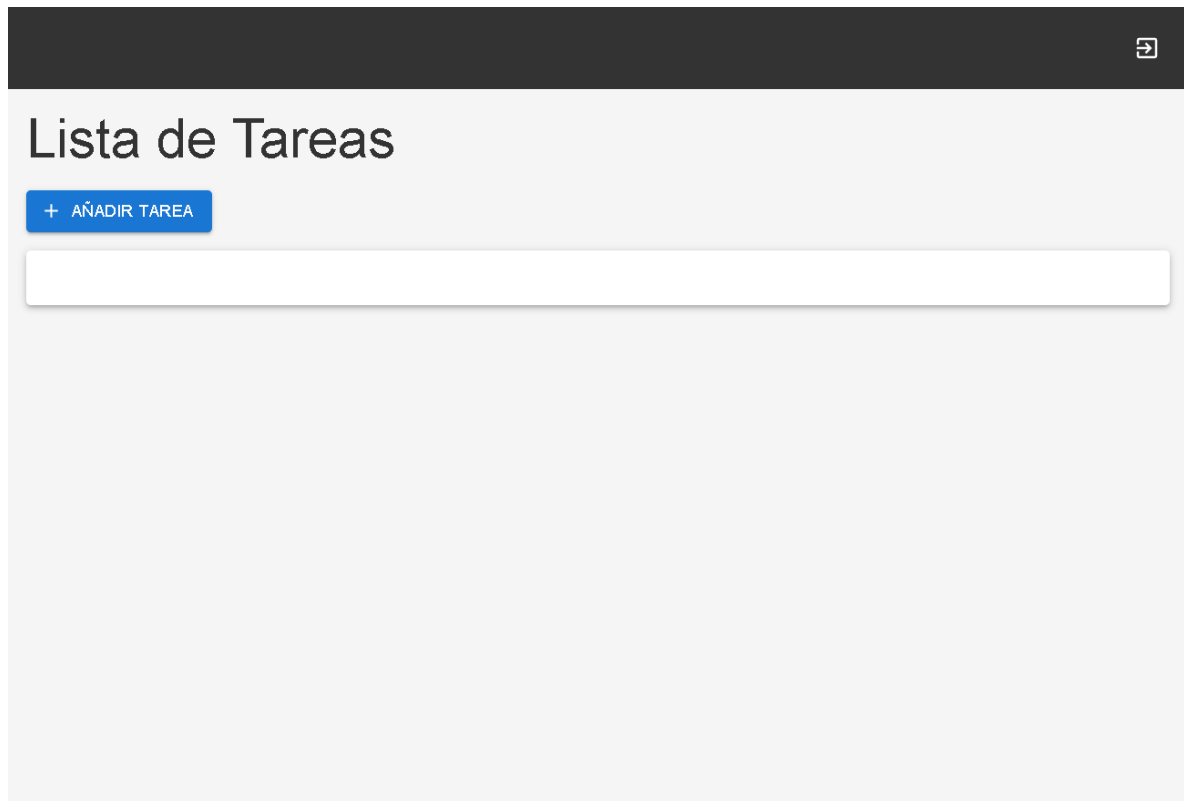


Ilustración 16 Captura de la parte del resumen de los escenarios.

En código, se guardan capturas de la corrida del cucumber

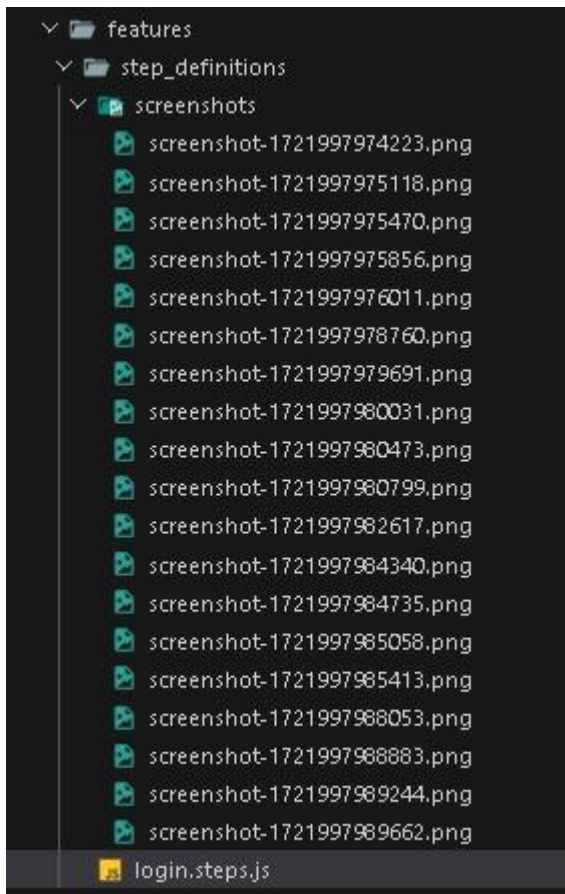


Ilustración 17 Screenshots en código

Conclusión

Las pruebas automatizadas del sistema de login desarrollado con React y Next.js, utilizando Cucumber y Selenium, han demostrado que el sistema es funcional y robusto en términos de autenticación de usuarios. Las pruebas de compatibilidad confirmaron que el sistema opera de manera consistente en diferentes navegadores y dispositivos. Además, las evaluaciones de seguridad indicaron que se implementaron medidas adecuadas para proteger los datos de los usuarios y prevenir ataques comunes.

También debemos acotar que debemos seguir usando estos software's, que nos facilitan y nos dan una mejor retroalimentación del producto que estamos brindando a un futuro cliente

Recomendaciones

- Mejorar la Gestión de Errores:**
 - Implementar mensajes de error más descriptivos y útiles para el usuario final, lo cual puede mejorar la experiencia de usuario y reducir la fricción durante el proceso de autenticación.
- Optimizar las Pruebas de Compatibilidad:**

- Continuar ejecutando pruebas regulares en una variedad de navegadores y dispositivos, especialmente a medida que se lanzan nuevas versiones y dispositivos.

3. **Fortalecer la Seguridad:**

- Realizar auditorías de seguridad periódicas y considerar la implementación de autenticación de dos factores (2FA) para una capa adicional de protección.