

Testing


Introducción breve: Descripción general de la aplicación.

Resumen de los tests, Tabla o listado con:

- Nombre del integrante
- Tipo de prueba realizada
- Descripción breve del componente probado
- Herramienta o framework usado
- Screenshot del código del test
- Resultado de la ejecución: Screenshot del test ejecutado y su salida.

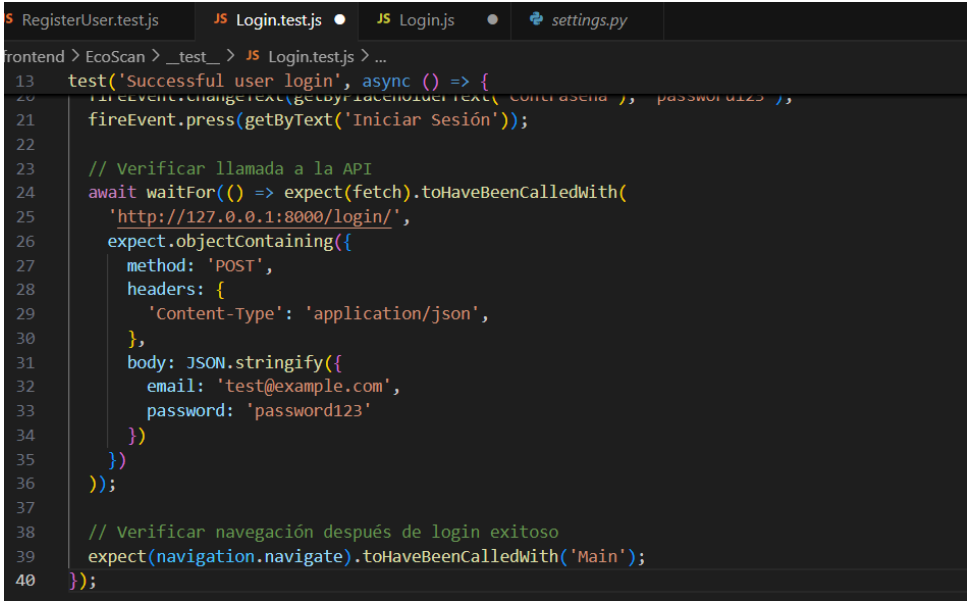
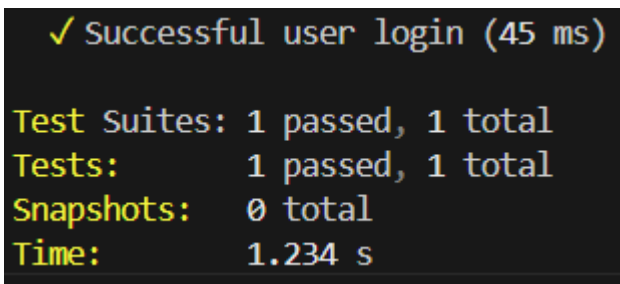
Lecciones aprendidas y dificultades: Reflexión grupal sobre la experiencia de testear.

EcoScan es una aplicación móvil diseñada para mejorar la gestión de residuos y el reciclaje utilizando inteligencia artificial. Su objetivo principal es permitir a los usuarios escanear objetos con la cámara de su dispositivo móvil para que puedan identificar su tipo de material y recibir recomendaciones sobre cómo desecharlos correctamente.

Nombre	Daniel Santiago Delgado Pinilla
Prueba	Prueba de integración
Descripción	Prueba de conexión entre el frontend (React Native)
Herramienta	Jest
Screenshot	 A screenshot of a code editor showing Jest test code for RegisterUser.test.js. The code includes imports for React, testing-library/react-native, and RegistroScreen. It defines a mock global.fetch function and a test case 'Register a new user' that renders the RegistroScreen, simulates user input, and asserts that the fetch function was called with the correct URL and method.

Resultado	<pre> Watch Usage > Press f to run only failed tests. > Press o to only run tests related to changed files. > Press p to filter by a filename regex pattern. > Press t to filter by a test name regex pattern. > Press q to quit watch mode. > Press Enter to trigger a test run. console.log CSRF token recibido at log (app/register.js:15:28) PASS __test__/RegisterUser.test.js ✓ Register a new user (316 ms) Test Suites: 1 passed, 1 total Tests: 1 passed, 1 total Snapshots: 0 total Time: 1.638 s, estimated 2 s Ran all test suites. Watch Usage: Press w to show more. </pre>
-----------	---

Nombre	Alejandro Argüello Muñoz
Prueba	Prueba de Login
Descripción	Prueba de logueo de un usuario en la aplicación
Herramienta	Jest

Screenshot	 <pre> 5 RegisterUser.test.js JS Login.test.js JS Login.js settings.py frontend > EcoScan > _test_ > JS Login.test.js > ... 13 test('Successful user login', async () => { 20 fireEvent.changeText(getByPlaceholderText('Contraseña'), 'password123'); 21 fireEvent.press(getByText('Iniciar Sesión')); 22 23 // Verificar llamada a la API 24 await waitFor(() => expect(fetch).toHaveBeenCalled()); 25 expect(fetch).toHaveBeenCalledWith(26 'http://127.0.0.1:8000/login/', 27 expect.objectContaining({ 28 method: 'POST', 29 headers: { 30 'Content-Type': 'application/json', 31 }, 32 body: JSON.stringify({ 33 email: 'test@example.com', 34 password: 'password123' 35 }) 36 }) 37); 38 // Verificar navegación después de login exitoso 39 expect(navigation.navigate).toHaveBeenCalledWith('Main'); 40 }); </pre>
Resultado	 <pre> ✓ Successful user login (45 ms) Test Suites: 1 passed, 1 total Tests: 1 passed, 1 total Snapshots: 0 total Time: 1.234 s </pre>

Lecciones aprendidas:

- Nos dimos cuenta de que hacer las pruebas desde el inicio facilita las cosas pues es más fácil darse cuenta de errores en las fases tempranas de desarrollo. Por ejemplo, en el registro, ocurrió un error del que no me había dado cuenta y había seguido programando. Ya después cuando se añadan más cosas va a ser más complicado arreglar los problemas.
- Hacer las pruebas desde el inicio facilita las cosas pues ya que es más fácil darse cuenta de los errores al inicio de desarrollo. Sin embargo, configurar el test fue una tarea complicada.