**Aplicación seleccionada:**

* Nombre de la aplicación: WebDriverIO demo (wdio)
* Versión de la aplicación: 1.0.8
* Fuente de descarga: https://github.com/webdriverio/native-demo-app/releases
* Descripción general: Aplicación móvil diseñada específicamente para fines de aprendizaje y pruebas de automatización.

**Preparación y Entorno de Pruebas**

* Herramientas utilizadas: Appium, Appium Inspector, Android Studio (dispositivo emulado), JUnit, Selenium WebDriver, Java (JDK 21), Maven, Android SDK, UiAutomator2, ADB.
* Configuración del entorno:

 Java Development Kit (JDK) 21

 Android SDK (dispositivo emulado Pixel 7)

 Actualización de Appium e instalación de Appium inspector (Host: 127.0.0.1 – Puerto 4723 – Path: /)

 Proyecto Maven con dependencias (Ver pom.xml)

 Instalación de la aplicación WDIO Demo (apk)

**Objetivos de las pruebas**

* Se validó que la aplicación inicie correctamente y que se pueda navegar entre las secciones principales como “Login”, “Forms”, y “Swipe”. También se probó el cierre de la aplicación.
* Se automatizaron interacciones frecuentes (iniciarSesion, inputField, botonFormulario, swipeHorizontal)
* Se verificó que los campos de email y contraseña estén visibles y habilitados para su uso. También se validó que el texto ingresado en el formulario se conserve y muestre correctamente
* Cada test incluye Assertions para validar el comportamiento esperado, lo cuál ayuda a detectar cualquier error si un elemento no aparece, si no se puede interactuar con él o si el resultado no es el esperado.

Escenarios de Prueba

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **N°** | **Descripción del escenario** | **Tipo de interacción** | **Resultado esperado** | **¿Funciona? (✓ ó ✗)** | **Observaciones** |
| **1** | La app inicia correctamente | Inicio de aplicación | La app se abre sin errores y muestra la pantalla principal | ✓ | Verificado en el método verificarInicio |
| **2** | Login exitoso con email y contraseña | Ingreso de datos + botón | Aparece el mensaje “You are logged in!” | ✓ | Validado en iniciarSesion |
| **3** | Campo de email visible y habilitado | Validación visual + estado de campo | El campo de email está presente y es editable | ✓ | Cubierto por validateEmailField |
| **4** | Campo de contraseña visible y habilitado | Validación visual + estado de campo | El campo de password está presente y es editable | ✓ | Cubierto por validatePasswordField |
| **5** | Ingreso de texto en formulario y validación del contenido | Entrada de texto | |  | | --- | |  |  |  | | --- | | El texto ingresado se mantiene y puede ser verificado | | ✓ | Ejecutado en inputField |
| **6** | Botón del formulario es clickeable | Clic de botón | Se puede presionar el botón | ✓ | |  | | --- | |  |  |  | | --- | | Testeado en botonFormulario | |
| **7** | Mensaje mostrado tras enviar formulario | Verificación post-acción | Se muestra un mensaje de confirmación (ej. "This button is active") | ✓ | Incluido en botonFormulario |
| **8** | Swipe horizontal hasta encontrar texto específico | Swipe (deslizamiento) | Se visualiza el texto "COMPATIBLE" tras el swipe | ✓ | Validado en swipeHorizontal |
| **9** | App se cierra correctamente | Cierre de aplicación | La app se cierra sin errores | ✓ | Verificado en cerrarApp |
| **10** | Validación de flujo completo de login + navegación | Secuencia de interacciones | Desde inicio hasta login, todo el flujo se ejecuta sin fallas | ✓ | Cubierto principalmente por iniciarSesion, en combinación con verificarInicio |

**Reflexión final**

* ¿Qué funcionó bien durante las pruebas?  
  Las pruebas automatizadas funcionaron correctamente en su mayoría. Se logró verificar que la aplicación inicia correctamente, acepta credenciales válidas, responde a la interacción con campos de texto y botones, y permite realizar acciones como swipe y cierre de forma efectiva. También se validó que los elementos estaban visibles y habilitados en el momento esperado.
* ¿Qué aspectos podrías mejorar en tu código?  
  Podría mejorar el uso de constantes reutilizables para evitar repetir localizadores (por ejemplo, los íconos con texto "󰍂", "󰏫", etc.), y separar la lógica de acciones en métodos más pequeños y reutilizables. También, a futuro, puedo aplicar el patrón Page Object Model para organizar mejor el código y facilitar su mantenimiento.
* ¿Qué aprendiste al realizar esta actividad?  
  Aprendí a conectar Appium con un emulador Android, automatizar escenarios comunes de prueba y a validar comportamientos esperados mediante aserciones. Además, entendí la importancia de la sincronización en pruebas móviles (usando WebDriverWait) y de tener un orden lógico y funcional en los test. Esta experiencia me ayudó a afianzar la estructura de pruebas automatizadas y a aplicar buenas prácticas desde el principio.