

Plan de Pruebas - Contact List Application

1. Resumen

Framework de automatización para Contact List App usando Playwright + Cucumber con JavaScript. Cubre pruebas de UI y API REST con enfoque en Page Object Model.

Resultados actuales: - 19 casos de prueba automatizados - 100% de casos pasando -
Tiempo de ejecución: ~3 minutos - CI/CD integrado con GitHub Actions

2. Alcance

- Autenticación (registro, login, logout)
 - CRUD de contactos
 - Validaciones de campos requeridos
 - API REST completa
-

3. Estrategia

Stack tecnológico

- **Playwright:** Auto-wait, soporte UI + API
- **Cucumber:** BDD con Gherkin
- **JavaScript:** Ecosistema moderno, fácil CI/CD

Patrón de diseño

Page Object Model con herencia: - BasePage con métodos comunes - 5 Page Objects específicos (Login, Signup, ContactList, AddContact, ContactDetails) - Selectores centralizados por página

Distribución de pruebas

- 60% API (rápidas, estables)
 - 30% UI (flujos críticos)
 - 10% E2E (smoke tests)
-

4. Casos de Prueba - UI

ID	Título	Precondiciones	Pasos	Resultado Esperado	Estado
UI-001	Registro de	Navegador en página principal	1. Clic en Sign	Usuario creado, redirige a /contactList	PASS

ID	Título	Precondiciones	Pasos	Resultado Esperado	Estado
	nuevo usuario		Up2. Llenar formulario (nombre, apellido, email único, contraseña) 3. Clic en Submit		
UI-002	Login con credenciales válidas	Usuario registrado previamente	1. Ingresar email 2. Ingresar contraseña 3. Clic en Submit	Login exitoso, redirige a /contactList	PASS
UI-003	Logout de sesión	Usuario autenticado	1. Clic en botón Logout	Sesión cerrada, redirige a login	PASS
UI-004	Crear contacto completo	Usuario autenticado	1. Clic en "Add a New Contact" 2. Llenar 11 campos	Contacto creado, visible en lista, navega a detalles	PASS

ID	Título	Precondiciones	Pasos	Resultado Esperado	Estado
			(nombre, apellido, fecha, email, teléfono, dirección, ciudad, estado, CP, país) ³ . Clic en Submit		
UI-005	Editar contacto	Usuario autenticado, contacto existente	1. Clic en contacto2. Clic en "Edit Contact"3. Modificar teléfono4. Clic en Submit	Teléfono actualizado, cambio reflejado en detalles	PASS
UI-006	Eliminar contacto	Usuario autenticado, contacto existente	1. Clic en contacto2. Clic en "Delete Contact"3. Acept	Contacto eliminado, no visible en lista	PASS

ID	Título	Precondiciones	Pasos	Resultado Esperado	Estado
			ar diálogo de confir mació n		
UI-007	Validación campo requerido	Usuario autenticado	1. Clic en "Add a New Contact"2. Llenar solo apellido3. Clic en Submit	Error: "firstName is required", contacto no creado	PASS

5. Casos de Prueba - API

ID	Endpoint	Método	Título	Payload	Status Esperado	Validación	Estado
API-001	/users	POST	Registrar usuario	{ "firstName": "Test", "lastName": "User", "email": "test@example.com", "password": "Pass123"}	201	Retorna token y user	PASS

ID	Endpoint	Método	Título	Payload	Status Esperado	Validación	Estado
API-002	/users/login	POST	Login exitoso	{"email": "test@example.com", "password": "Pass123"}	200	Retorna token válido	PASS
API-003	/users/login	POST	Login inválido	{"email": "wrong@example.com", "password": "Wrong"}	401	Retorna mensaje de error	PASS
API-004	/users/me	GET	Obtener perfil	Header: Authorization: Bearer {token}	200	Retorna datos de usuario	PASS
API-005	/users/me	PATCH	Actualizar perfil	{"firstName": "Updated"}	200	firstName actualizado	PASS
API-006	/users/me	DELETE	Eliminar usuario	Header: Authorization: Bearer {token}	200	Usuario eliminado	PASS

ID	Endpoint	Método	Título	Payload	Status Esperado	Validación	Estado
API-007	/contacts	POST	Crear contacto	{ "firstName": "Juan", "lastName": "Pérez", "email": "juan@email.com" }	201	Retorna contacto con _id	PASS
API-008	/contacts	GET	Listar contactos	Header: Authorization: Bearer {token}	200	Array de contactos	PASS
API-009	/contacts/:id	GET	Obtener contacto	Header: Authorization: Bearer {token}	200	Datos del contacto	PASS
API-010	/contacts/:id	PATCH	Actualizar contacto	{ "phone": "3109999999" }	200	Teléfono actualizado	PASS
API-011	/contacts/:id	DELETE	Eliminar contacto	Header: Authorization: Bearer {token}	200	Contacto eliminado	PASS

ID	Endpoint	Método	Título	Payload	Status Esperado	Validación	Estado
API-012	/contacts	POST	Validación sin firstName	{"lastName": "Test"}	400	Error de validación	PASS

6. Datos de Prueba

Estrategia

Generación dinámica con timestamps para evitar conflictos:

Email: test_1771012345678@example.com

Compartir datos

Uso del contexto de Cucumber (`this.emailRegistrado`, `this.passwordRegistrado`) para reutilizar credenciales entre pasos del mismo escenario.

7. Manejo de Estabilidad

Técnicas aplicadas

- Auto-wait de Playwright
- Timeout de 120 segundos en steps

Resultados

- 0% falsos positivos
- 100% reproducibilidad

8. CI/CD

Pipeline (GitHub Actions)

1. Install dependencies
2. Run UI tests (paralelo)
3. Run API tests (paralelo)
4. Generate HTML reports
5. Upload artifacts (screenshots, reportes)

Trigger

- Push a main/develop

- Pull requests
 - Ejecución manual
-

9. Riesgos

Riesgo	Probabilidad	Impacto	Mitigación
BD volátil	Alta	Medio	Timestamps únicos
Selectores cambian	Media	Alto	Page Object Model, fallback
Latencia en CI	Media	Medio	Timeouts generosos, reintentos

10. Métricas

Métrica	Objetivo	Actual
Cobertura de requisitos	>85%	100%
Casos automatizados	>15	19
Tiempo de ejecución	<5 min	3 min
Tasa de éxito	>95%	100%
Falsos positivos	<5%	0%

11. Estructura del Proyecto

funcionalidades/

```
├── pages/
│   ├── BasePage.js
│   ├── LoginPage.js
│   ├── SignupPage.js
│   ├── ContactListPage.js
│   ├── AddContactPage.js
│   └── ContactDetailsPage.js
├── definiciones_pasos/
│   ├── login_steps.js
│   ├── signup_steps.js
│   ├── contactos_steps.js
│   └── api_steps.js
└── soporte/
    └── world.js
```

12. Comandos de Ejecución

```
npm test           # Todas Las pruebas
npm run test:ui    # Solo UI
npm run test:api   # Solo API
npm run test:smoke # Smoke tests
npm run test:report # Ver reporte HTML
```

Anexo A: Ejemplo de Caso en Gherkin

Scenario: Crear un contacto nuevo con todos los datos
Given que el usuario está registrado y autenticado
When el usuario hace clic en "Add a New Contact"
And el usuario llena el formulario de contacto con:
firstName	Juan
lastName	Pérez
email	juan@email.com
And el usuario hace clic en el botón "Submit"
Then debería ver el contacto "Juan Pérez" en la lista
And debería poder ver los detalles del contacto

Anexo B: Matriz de Trazabilidad

Requisito	Descripción	Casos de Prueba	Estado
REQ-001	Usuario puede registrarse	UI-001, API-001	CUBIERTO
REQ-002	Usuario puede iniciar sesión	UI-002, API-002	CUBIERTO
REQ-003	Usuario puede cerrar sesión	UI-003	CUBIERTO
REQ-004	Usuario puede crear contacto	UI-004, API-007	CUBIERTO
REQ-005	Usuario puede editar contacto	UI-005, API-010	CUBIERTO
REQ-006	Usuario puede eliminar contacto	UI-006, API-011	CUBIERTO
REQ-007	Sistema valida campos requeridos	UI-007, API-012	CUBIERTO
REQ-008	API retorna 401 para auth inválida	API-003	CUBIERTO
REQ-009	API requiere token para	API-004, API-008	CUBIERTO

Requisito	Descripción	Casos de Prueba	Estado
	endpoints protegidos		
REQ-010	Usuario puede ver detalles de contacto	UI-004, API-009	CUBIERTO