

# DOCUMENTO PRUEBAS DE SOFTWARE 2

UNIVERSIDAD COMPENSAR

JUNNIOR A. HERNANDEZ JUAN DAVID LEON  
JUAN SEBASTIAN ACERO

SERGIO ALEXANDER MORA

15/04/2025

# Proyecto Final - Pruebas Automatizadas

## Descripción del Sistema Probado

El sistema probado es la aplicación web de demostración DemoQA (<https://demoqa.com>), una plataforma de práctica que ofrece diversos componentes y funcionalidades web para realizar pruebas. Esta aplicación contiene diferentes secciones como Elements, Forms, Alerts/Frames/Windows e Interactions, cada una con variados componentes web que permiten practicar diferentes escenarios de pruebas automatizadas.

## Objetivos de las Pruebas

Los objetivos principales de nuestras pruebas automatizadas son:

1. Verificar la funcionalidad correcta de los componentes principales en DemoQA
2. Validar la navegación entre diferentes secciones de la aplicación
3. Comprobar el funcionamiento de formularios e interacciones de usuario
4. Evaluar el comportamiento de elementos interactivos como alertas y elementos arrastables
5. Documentar cualquier comportamiento inesperado o error encontrado durante las pruebas

## Tabla de Casos de Prueba

### Caso de Prueba 1: PruebaElements

**Objetivo:** Verificar la funcionalidad de la sección "Elements" y el formulario de texto.

ID	Descripción	Pasos	Resultado Esperado	Resultado Obtenido
TC-01	Verificar formulario de texto en la sección "Elements"	1. Visitar la página principal 2. Hacer clic en la sección "Elements" 3. Seleccionar "Text Box" 4. Completar los campos del formulario 5. Enviar el formulario	Los datos ingresados deben mostrarse en la parte inferior del formulario después de enviar	PASÓ - Los datos se mostraron correctamente

## Caso de Prueba 2: PruebaForms

**Objetivo:** Verificar la funcionalidad de los formularios de registro.

ID	Descripción	Pasos	Resultado Esperado	Resultado Obtenido
TC-02	Completar formulario de práctica	1. Visitar la página principal 2. Navegar a la sección "Forms" 3. Seleccionar "Practice Form" 4. Completar campos obligatorios (nombre, apellido, número, dirección) 5. Seleccionar género 6. Marcar hobbies 7. Enviar el formulario	El formulario debe enviarse correctamente y mostrar una confirmación	PASÓ - El formulario se envió correctamente

## Caso de Prueba 3: PruebaAlerts

**Objetivo:** Verificar la funcionalidad de alertas, ventanas y marcos.

ID	Descripción	Pasos	Resultado Esperado	Resultado Obtenido
TC-03	Probar funcionalidad de browser windows y frames	1. Visitar la página principal 2. Navegar a la sección "Alerts, Frame & Windows" 3. Probar los diferentes tipos de ventanas (tab, window, message) 4. Navegar a la sección de frames 5. Interactuar con los frames	Las nuevas ventanas/pestañas deben abrirse correctamente y los frames deben ser accesibles	PASÓ - Las ventanas y frames funcionaron según lo esperado

## Caso de Prueba 4: PruebaInteractions

**Objetivo:** Verificar la funcionalidad de interacciones de usuario como arrastrar y soltar.

ID	Descripción	Pasos	Resultado Esperado	Resultado Obtenido
TC-04	Probar funcionalidades de redimensionamiento y arrastre	1. Visitar la página principal 2. Navegar a la sección "Interactions" 3. Probar las funcionalidades de "Resizable" 4. Probar las diferentes opciones de "Draggable" 5. Interactuar con los elementos arrastrables	Los elementos deben poder redimensionarse y arrastrarse según las restricciones especificadas	PASÓ - Los elementos se redimensionaron y arrastraron correctamente

## Observaciones y Hallazgos

Durante la ejecución de las pruebas automatizadas, observamos los siguientes aspectos:

1. **Navegación fluida:** La navegación entre secciones de la aplicación funcionó sin problemas, permitiendo acceder a todas las funcionalidades probadas.
2. **Formularios funcionales:** Los formularios en la aplicación aceptaron correctamente los datos ingresados y respondieron adecuadamente a la acción de envío.
3. **Interacciones complejas:** Las funcionalidades de arrastrar y soltar, así como el redimensionamiento de elementos, requirieron script de pruebas más elaborados pero funcionaron según lo esperado.
4. **Alertas y ventanas:** La aplicación manejó correctamente la apertura de nuevas ventanas y pestañas del navegador.
5. **Áreas de mejora en las pruebas:**
  - Se podrían implementar validaciones más explícitas para verificar el estado después de cada acción
  - Falta manejo de alertas en las pruebas (actualmente solo se disparan)
  - Sería beneficioso añadir aserciones para validar el estado esperado después de las interacciones

## Recomendaciones de Mejora

Para mejorar la calidad y eficacia de nuestras pruebas automatizadas, recomendamos:

1. **Implementar aserciones explícitas:** Añadir verificaciones después de cada acción para confirmar que el sistema respondió como se esperaba.
2. **Mejorar el manejo de alertas:** Implementar código para confirmar, cancelar o ingresar texto en alertas según sea necesario.
3. **Parametrizar datos de prueba:** Extraer los datos de prueba a archivos externos (JSON o CSV) para facilitar la ejecución con diferentes conjuntos de datos.
4. **Implementar Page Objects:** Refactorizar las pruebas utilizando el patrón Page Object Model para mejorar la mantenibilidad y reutilización.

5. **Añadir pruebas negativas:** Incluir escenarios donde se espera que el sistema maneje errores (por ejemplo, formularios con datos inválidos).
6. **Mejorar la documentación del código:** Añadir comentarios descriptivos a las pruebas para facilitar su comprensión y mantenimiento.

## Instrucciones para Ejecutar las Pruebas

Para ejecutar estas pruebas en un entorno local:

1. Clonar el repositorio: `git clone [URL del repositorio]`
2. Instalar dependencias: `npm install`
3. Iniciar Cypress: `npx cypress open`
4. Seleccionar el navegador deseado
5. Hacer clic en los archivos de prueba para ejecutarlos individualmente, o ejecutar todos con `npx cypress run`

## Requisitos del Sistema

- Node.js (versión 12 o superior)
- Cypress (versión 12.0.0 o superior)
- Navegador web compatible (Chrome, Firefox, Edge)

## Conclusiones

Las pruebas automatizadas implementadas en Cypress para la aplicación DemoQA demostraron ser efectivas para validar las principales funcionalidades del sistema. Pudimos verificar con éxito la interacción con elementos, formularios, alertas e interacciones complejas como arrastrar y soltar.

El framework Cypress proporcionó una plataforma robusta para implementar estas pruebas, ofreciendo buena visibilidad y facilidad de uso. Sin embargo, identificamos oportunidades de

mejora para hacer nuestras pruebas más completas y mantenibles en el futuro.

Este proyecto nos permitió aplicar los conocimientos adquiridos sobre pruebas automatizadas en un entorno real, enfrentando desafíos prácticos y encontrando soluciones efectivas para validar el funcionamiento de aplicaciones web modernas.