Actividad Unidad 2

Liborio De Jesús Castañeda Valencia

Prof. Reynel Alfonso Mendoza Garrido

Universidad De Cartagena

Pruebas Y Mantenimiento De Software

Cartagena, Bolívar

2025

# Informe de Pruebas de Software

## 1. Revisión del programa

El programa seleccionado fue el proyecto “Información Personal” disponible en GitHub:  
**Enlace público**: <https://liboriocastaneda99.github.io/informacion-personal/>

**Enlace repositorio**: <https://github.com/LiborioCastaneda99/informacion-personal>   
  
**Descripción**:  
Es una aplicación web sencilla que permite registrar datos personales como nombre, apellidos, documento, correo, teléfono, dirección, entre otros. El sistema muestra los registros en forma de tarjetas en la misma interfaz.  
  
**Componentes evaluados**:  
- Formulario de registro: incluye validaciones de campos obligatorios y formatos correctos.  
- Módulo de visualización: genera tarjetas con los datos registrados.  
- Persistencia en memoria (front-end): mantiene registros en la misma sesión del navegador.

## 2. Diseño de casos de prueba

|  |  |  |  |
| --- | --- | --- | --- |
| Caso | Descripción | Entrada | Resultado esperado |
| CP01 | Registro válido | Todos los campos completos y correctos | Se crea tarjeta |
| CP02 | Nombre vacío | Nombre vacío, demás campos correctos | No debe crearse tarjeta |
| CP03 | Correo inválido | Correo con formato incorrecto | No debe crearse tarjeta |
| CP04 | Documento con letras | Ingresar letras en campo documento | No debe crearse tarjeta |
| CP05 | Teléfono vacío | Teléfono vacío, demás campos correctos | No debe crearse tarjeta |
| CP06 | Múltiples registros | Ingresar 3 registros válidos distintos | Deben aparecer al menos 3 tarjetas |

## 3. Ejecución de las pruebas

Se ejecutaron los casos de prueba de forma automática con Selenium WebDriver (Python). El script abre el navegador, llena los formularios según cada escenario y valida los resultados. Al finalizar, genera un reporte en Word con una tabla que resume los resultados.  
  
**Evidencias**:  
- Se adjuntan capturas de pantalla generadas automáticamente en cada caso de prueba.  
- Se incluye el archivo 'reporte\_pruebas.docx' con la tabla de resultados.  
  
Ejemplo de resultado parcial en consola:  
✅ CP01 OK  
✅ CP02 OK  
✅ CP03 OK  
✅ CP04 OK  
✅ CP05 OK  
✅ CP06 OK

## 4. Informe de resultados

Resumen:

- Total de casos ejecutados: 6

- Casos exitosos: 6

- Casos fallidos: 0

**Conclusiones**:

Comprensión del ciclo de pruebas: La actividad permitió aplicar de manera práctica las fases fundamentales del proceso de pruebas de software: revisión del programa, diseño de casos de prueba, ejecución y documentación de resultados.

Importancia de los casos de prueba: Se evidenció que un buen diseño de casos de prueba ayuda a cubrir diferentes escenarios (positivos, negativos y alternativos), lo que aumenta la probabilidad de detectar defectos tempranamente.

Uso de herramientas de prueba: El empleo de Selenium para la automatización y python-docx para la documentación demostró cómo las herramientas apoyan el aseguramiento de la calidad y agilizan el proceso de validación.

Detección de errores y mejora continua: Durante la ejecución se identificaron errores en el flujo de la aplicación, lo cual refuerza la idea de que las pruebas no solo buscan encontrar fallos, sino también mejorar la calidad del software mediante retroalimentación al desarrollo.

Aprendizaje integral: La práctica integró teoría y práctica, mostrando cómo aplicar los conceptos de diseño de pruebas en un entorno real y evidenciando la importancia de documentar los hallazgos en un informe claro y profesional.