Realizado por:
Santiago Gallardo
David Pérez
Isabela Zambrano

VENTAJAS Y DESVENTAJAS

PRUEBAS CAJA NEGRA Y BLANCA

Tabla Caja Negra

CAJA BLANCA	
Ventajas	**Desventajas**
Permiten analizar los caminos lógicos internos del código y asegurar su correcto funcionamiento. En el proyecto, se utilizaron grafos de flujo y cálculos de complejidad ciclomática para cubrir todas las rutas de validación de credenciales, cambio de usuario y modificación de contraseña, garantizando que cada decisión interna se ejecute correctamente.	Requieren conocimientos técnicos avanzados sobre el código fuente para su aplicación en el proyecto, fue necesario que los desarrolladores comprendieran los métodos de conexión, validación y modificación, lo que limita la participación de testers no técnicos.
Ayudan a identificar errores en condiciones lógicas complejas estas pruebas fueron útiles para validar que el sistema actualizara correctamente los datos en la base MongoDB, verificando la lógica detrás de las operaciones críticas como la modificación segura de contraseñas.	Demandan mayor tiempo y esfuerzo en su elaboración la generación de diagramas de flujo, grafos, caminos básicos y complejidad ciclomática implica una inversión considerable de tiempo, especialmente cuando se cubren múltiples funcionalidades como contratos civiles y laborales.
Permiten evaluar funcionalidades críticas y sensibles del sistema funciones como el manejo de intentos de acceso, la seguridad en los cambios de datos del usuario, y la integración con la base de datos se sometieron a pruebas exhaustivas a nivel interno.	No evalúan la experiencia del usuario ni la estética de la interfaz, aunque el código funcione correctamente, este tipo de prueba no detecta si los mensajes son claros o si la disposición de la interfaz resulta comprensible para el usuario.
Favorecen la optimización del código y su mantenimiento gracias al análisis estructural, el equipo pudo mejorar la calidad del software, eliminando redundancias o condiciones innecesarias en el flujo de control.	Pueden omitir errores de integración entre lógica y presentación, por ejemplo, si el backend guarda correctamente el nuevo nombre de usuario, pero la interfaz no lo muestra actualizado, este error no sería detectado por pruebas internas del código.

Tabla Caja Blanca

CAJA NEGRA	
Ventajas	**Desventajas**
Permiten verificar el cumplimiento de los requisitos funcionales sin necesidad de conocer el código fuente en el proyecto, estas pruebas permitieron evaluar funcionalidades clave como el inicio de sesión, la generación de contratos y la información del administrador.	No permiten identificar errores lógicos internos del sistema, aunque se verificó que el bloqueo del login funcionara después de 3 intentos fallidos, estas pruebas no garantizan que el contador de intentos esté funcionando correctamente a nivel de código.
Facilitan la simulación de escenarios de uso real mediante entradas válidas e inválidas el equipo realizó pruebas de equivalencia para verificar distintos casos como contraseñas incorrectas, confirmaciones erróneas y cambios exitosos de usuario.	La cobertura del código fuente es limitada las condiciones internas del método de validación con MongoDB o los procedimientos que actualizan los datos del usuario podrían contener errores que no se detectan desde la interfaz.

CAJA NEGRA	
Ventajas	**Desventajas**
Son adecuadas para validar la interfaz gráfica del sistema se aplicaron con éxito para comprobar que las pantallas como “Login”, “Gestión de contratos”, “Contrato Civil” o “Contrato Laboral” respondan correctamente a los botones y entradas del usuario.	No detectan la existencia de código muerto o caminos no ejecutados si alguna condición lógica en el código nunca se activa, este tipo de prueba no la detectará, ya que se enfoca exclusivamente en los resultados visibles.
Detectan fallos visibles para el usuario final por ejemplo permitieron verificar la aparición de mensajes de error como “usuario o contraseña no válido” o comprobar si se muestra el formulario correcto al seleccionar un tipo de contrato.	Dependencia de una correcta definición de los casos funcionales si no se contemplan todos los casos de entrada en las tablas de clases equivalentes, podrían quedar escenarios sin probar, lo que comprometería la calidad del sistema.

Bibliografía

Ruiz, J. y Lucio, X. (2025) Fundamentos de ingeniería de software. Universidad de las Fuerzas Armadas ESPE