

Contenido

[INTRODUCCION 3](#_Toc183455654)

[***¿Qué es una prueba de software?*** 3](#_Toc183455655)

[***¿Para qué sirven las pruebas de software?*** 3](#_Toc183455656)

[PROYECTO 4](#_Toc183455657)

[***Descripción*** 4](#_Toc183455658)

[OBJETIVO 5](#_Toc183455659)

[PRUEBAS DE SOFTWARE 6](#_Toc183455660)

[*Módulo Principal:* 6](#_Toc183455661)

[Prueba de Fiabilidad 6](#_Toc183455662)

[Prueba de Usabilidad 7](#_Toc183455663)

[Prueba de Funcionabilidad 8](#_Toc183455664)

[*Módulo de Inicio de Sesión:* 9](#_Toc183455665)

[Prueba de Caja blanca 9](#_Toc183455666)

[Prueba de Caja negra 11](#_Toc183455667)

[Prueba de Usabilidad 12](#_Toc183455668)

[*Módulo de Registro:* 13](#_Toc183455669)

[Prueba Caja negra 13](#_Toc183455670)

[Prueba de Fiabilidad 14](#_Toc183455671)

[Mantenibilidad 15](#_Toc183455672)

[*Módulo de casos Actuales:* 16](#_Toc183455673)

[Submódulo de “Visualización de todos los casos” 16](#_Toc183455674)

# INTRODUCCION

Las pruebas de software son una parte esencial del proceso de desarrollo de aplicaciones y sistemas informáticos. Estas pruebas buscan asegurar que el software cumpla con los requisitos definidos, funcione de manera adecuada y esté libre de defectos antes de ser lanzado al usuario final. En un mundo donde las aplicaciones y sistemas informáticos son cada vez más complejos, el proceso de prueba se ha vuelto indispensable para garantizar la calidad del software y minimizar los riesgos asociados con su funcionamiento incorrecto.

## ***¿Qué es una prueba de software?***

Una prueba de software es una actividad realizada con el objetivo de verificar y validar que un sistema de software esté funcionando correctamente según lo esperado. Las pruebas buscan identificar defectos, inconsistencias y vulnerabilidades en el software, y al mismo tiempo, asegurar que el software cumpla con las especificaciones y los requisitos del cliente o usuario final. A través de las pruebas, se valida que el producto final no solo sea funcional, sino también seguro, confiable y fácil de usar.

Las pruebas de software pueden llevarse a cabo de diferentes formas, dependiendo de la etapa del ciclo de vida del desarrollo en la que se encuentren. Existen enfoques manuales, en los que un tester realiza las verificaciones, y enfoques automatizados, en los que herramientas especializadas se encargan de ejecutar las pruebas de manera repetitiva y controlada.

## ***¿Para qué sirven las pruebas de software?***

Las pruebas de software cumplen un rol crucial al garantizar que el producto final sea confiable, funcional y de alta calidad. Sirven para:

* ***Detectar y corregir errores:*** Las pruebas ayudan a identificar defectos en el código y en el diseño, lo que permite corregirlos antes de que el software llegue al usuario final.
* ***Validar los requisitos:*** Permiten verificar que el sistema cumpla con los requisitos especificados por los usuarios o el cliente, asegurando que el producto entregado sea el que se esperaba.
* ***Asegurar la calidad del producto:*** Las pruebas son fundamentales para asegurar que el software sea estable, eficiente y fácil de usar.
* ***Reducir el riesgo de fallos:*** Al identificar problemas de manera temprana, las pruebas ayudan a mitigar los riesgos de que el software falle en producción, lo que podría generar pérdidas económicas y de reputación

Todo esto para evitar problemas durante el desarrollo y/o futuros como, por ejemplo:

# PROYECTO

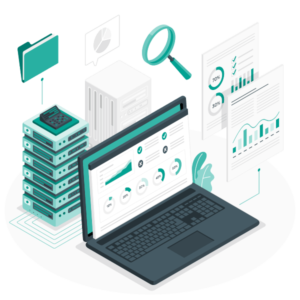
El proyecto tendrá el nombre de “LexSolution” desarrollado por la empresa EcoByte.

## ***Descripción***

Proporcionar a los abogados una aplicación que permita realizar su trabajo de manera más eficiente, reduciendo el uso de papel, digitalizando sus expedientes y casos, facilitando la gestión de alertas de calendario y cobros, y ofreciendo estadísticas que les ayuden a medir su productividad y crecimiento, ya sea como abogados independientes o dentro de una firma.

# OBJETIVO

El objetivo principal de las pruebas de software en este proyecto es identificar tanto fallos como aciertos durante las diferentes fases de su desarrollo, con el fin de poder corregir los errores detectados y continuar con el avance del proyecto de manera eficiente. Estas pruebas se realizarán de forma continua y progresiva, a medida que el software se va desarrollando, para asegurar que todas las funcionalidades y características del sistema se ajusten a las especificaciones previamente definidas y respondan adecuadamente a las expectativas del usuario final.



El proceso de prueba tiene como fin asegurar la calidad del producto, lo que implica verificar su rendimiento, funcionalidad, seguridad y usabilidad. Esto se logra mediante la identificación temprana de defectos que puedan comprometer su integridad, eficiencia y estabilidad, y la implementación de correcciones antes de que el software sea lanzado a producción. Asimismo, se busca confirmar que el software cumpla con todos los requisitos y estándares establecidos, minimizando los riesgos de fallos en un entorno de producción y garantizando que el sistema sea confiable y eficiente.

En el desarrollo de este proyecto, las pruebas se realizarán de manera estructurada, abarcando diferentes tipos de pruebas como pruebas unitarias, de integración, de sistema y de aceptación, dependiendo de la fase en la que se encuentre el desarrollo. El objetivo final es concluir con una evaluación exhaustiva y bien documentada del software, lo que permitirá asegurar un producto final de alta calidad que no solo cumpla con los requisitos funcionales, sino que también sea seguro, escalable y de fácil uso, con el fin de ofrecer una experiencia satisfactoria al usuario final y evitar problemas futuros que puedan afectar su desempeño y reputación.

# PRUEBAS DE SOFTWARE

## *Módulo Principal:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Prueba de Fiabilidad | | | | | |
|  | ***Modulo Principal*** | ***Versión 1*** | ***Tester***: Medellín Redondo Jorge Daniel | | **Fecha de Prueba: 22/11/2024** |
| ***Resultado*** ***esperado***: Manipulación sin interrupciones durante su uso por un tiempo prolongado | | | | | |
| ***Tópico a Evaluar*** | ***Bueno*** | ***Medio*** | ***Aceptable*** | ***Malo*** | ***Observaciones*** |
| El sistema no presenta caídas o interrupciones durante su uso. |  |  |  |  |  |
| El sistema mantiene la integridad de los datos sin inconsistencias. |  |  |  |  |  |
| Tiempo de respuesta aceptable en todas las operaciones |  |  |  |  |  |
| El sistema mantiene su integridad si el tamaño de la página cambia |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prueba de Usabilidad | | | | | | | | | | |
|  | | ***Modulo Principal*** | | ***Versión 1*** | | ***Tester***: Arriaga García Gael | | | | **Fecha de Prueba: 22/11/2024** |
| ***Resultado*** ***esperado***: El sistema debe proporcionar una interfaz intuitiva, fácil de navegar y comprender para el usuario promedio. Los usuarios deben ser capaces de realizar tareas comunes sin confusión, identificar claramente las opciones disponibles, y recibir mensajes de error claros y útiles en caso de problemas. | | | | | | | | | | |
| ***Tópico a Evaluar*** | ***Argumento*** | | ***Bueno*** | | ***Medio*** | | ***Aceptable*** | ***Malo*** | ***Observaciones*** | |
| Lenguaje | El lenguaje es fácil y entendible por el usuario | |  | |  | |  |  |  | |
| Visibilidad | El usuario es capaz de encontrar los botones o campos necesarios en un tiempo mínimo | |  | |  | |  |  |  | |
| Formato | Los colores del sistema no confunden, desorientan o hartan al usuario asimismo el formato de letra o diseño. | |  | |  | |  |  |  | |
| Intuitivo | El sistema permite al usuario entender las acciones a realizar de manera fácil y rápida | |  | |  | |  |  |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Prueba de Funcionabilidad | | | | | | |
|  | ***Modulo Principal*** | | ***Versión 1*** | ***Tester***: Guerra Ruiz Daniel | | **Fecha de Prueba: 22/11/2024** |
| ***Resultado*** ***esperado***: El sistema debe ejecutar todas las funcionalidades principales de manera correcta y sin errores bajo condiciones normales de uso. Cada módulo debe cumplir con los requerimientos establecidos, procesando datos válidos de forma precisa y rechazando datos inválidos con mensajes claros. | | | | | | |
| ***Tópico a Evaluar*** | | ***Bueno*** | ***Medio*** | ***Aceptable*** | ***Malo*** | ***Observaciones*** |
| Todas las funciones principales operan correctamente bajo condiciones normales. | |  |  |  |  |  |
| El sistema responde adecuadamente a datos válidos e inválidos. | |  |  |  |  |  |
| Integración correcta con otros módulos o componentes. | |  |  |  |  |  |

## *Módulo de Inicio de Sesión:*

***Introducción***: Para este módulo de Inicio de Sesión se intenta demostrara atreves de pruebas del sistema que se puede ingresar con la captura de los datos del usuario, es decir, su usuario y contraseña.

***Propósito***: Asegurar que la validación de los datos es correcta, que las cuentas no se combinen y no permitir el acceso de usuarios ajenos a la cuenta.

### Prueba de Caja blanca

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Módulo de Inicio de Sesión*** | ***Versión 1*** | ***Tester***: Medellín Redondo Jorge Daniel | **Fecha de Prueba: 23/11/2024** |

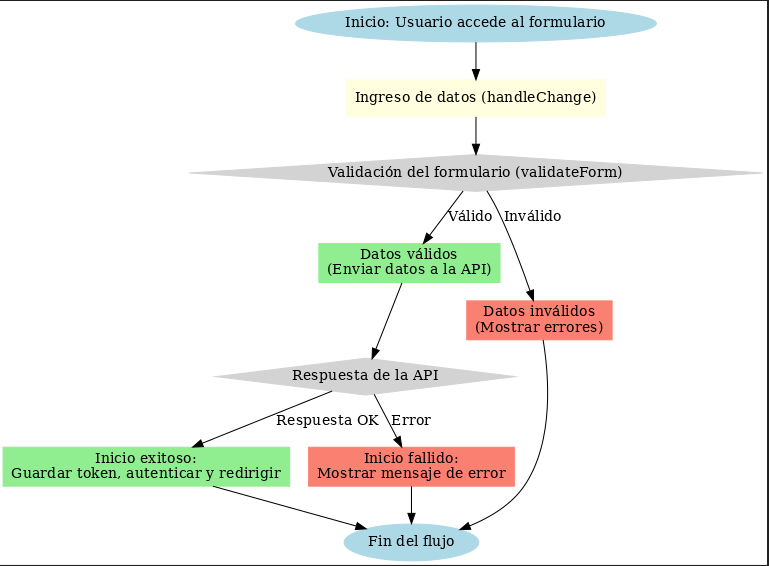


Ilustración Diagrama de Flujo de Inicio de Sesión

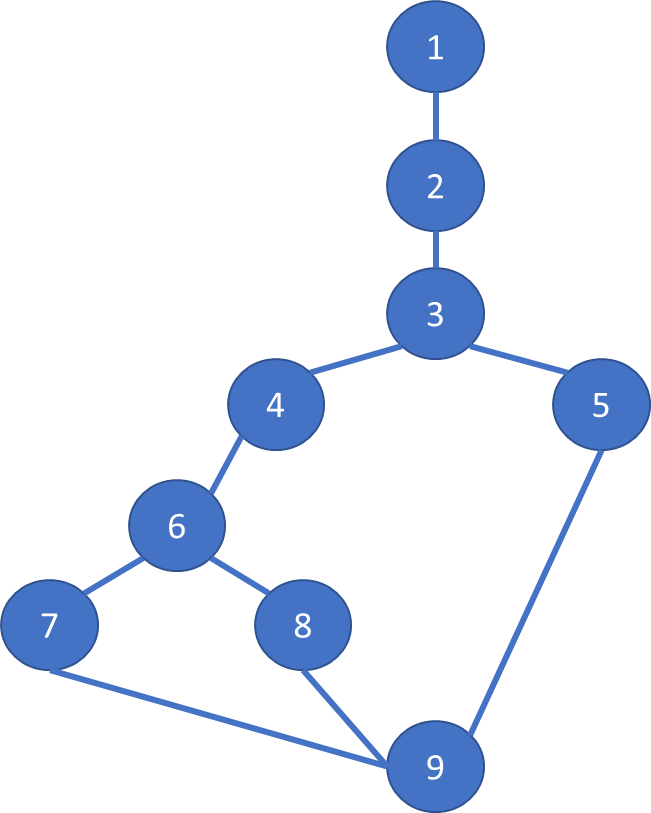
* Basado-grafo de Flujo:

Ilustración Grafo de Inicio de Sesión

V(G) = E – N + 2 = 10 – 9 + 2 = ***3***

* Basado código:

V(G) = 1 + D = 1 + 2 = ***3***

* Numero de Regiones:

V(G) = ***3***

* Caminos Básicos:

1. 1-3, 5, 9.
2. 1-3, 4, 6, 8, 9.
3. 1-3, 4, 6, 7, 9.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Objetivo | Datos de Entrada | Procedimiento | Salida esperada | Resultado |
| T1 | Verificar camino 1 | Nombre de Usuario  Contraseña de Usuario | Se validarán los datos del usuario, se ingresarán datos inválidos en ambos campos | Mensaje en Pantalla “Usuario no encontrado” |  |
| T2 | Verificar camino 2 | Nombre de Usuario  Contraseña de Usuario | Se validarán los datos del usuario, se ingresarán los datos y habrá un inicio fallido | Mensaje en Pantalla  “Inicio Fallido, revise los datos proporcionados” |  |
| T3 | Verificar camino 3 | Nombre de Usuario  Contraseña de Usuario | Se validarán los datos del usuario, se ingresarán los datos correctamente de un usuario ya registrado  Posteriormente se ira al módulo de Casos | Mensaje en Pantalla  “Bienvenido” |  |

### Prueba de Caja negra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Módulo de Inicio de Sesión*** | ***Versión 2*** | ***Tester***: Morales Gonzales Juan Pablo | **Fecha de Prueba: 23/11/2024** |

|  |  |  |
| --- | --- | --- |
| Datos de Entrada | Clases validas | Clases no validas |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Casos de Prueba | Clases validas | Clases no validas | Salida |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prueba de Usabilidad | | | | | | | | | | |
|  | | ***Módulo de Inicio de Sesión*** | | ***Versión 1*** | | ***Tester***: Medellín Redondo Jorge Daniel | | | | **Fecha de Prueba: 23/11/2024** |
| ***Resultado*** ***esperado***: Los mensajes deben ser claros y en el idioma del usuario. El flujo debe ser natural, guiando al usuario desde la entrada de datos hasta el resultado esperado. | | | | | | | | | | |
| ***Tópico a Evaluar*** | ***Argumento*** | | ***Bueno*** | | ***Medio*** | | ***Aceptable*** | ***Malo*** | ***Observaciones*** | |
| Lenguaje | Los mensajes de error y etiquetas deben ser comprensibles para el usuario. | |  | |  | |  |  |  | |
| Visibilidad | |  | | --- | | El usuario debe poder identificar los campos de entrada fácilmente. |  |  | | --- | |  | | |  | |  | |  |  |  | |
| Formato | Los colores y tipografía no deben confundir al usuario. | |  | |  | |  |  |  | |
| Intuitivo | El sistema debe permitir al usuario entender cómo interactuar sin ayuda externa. | |  | |  | |  |  |  | |
| Atajos | El usuario debe ser capaz de realizar acciones de forma eficiente. | |  | |  | |  |  |  | |
| Tiempo de respuesta | El tiempo entre el envío del formulario y la respuesta del servidor debe ser menor a 3 segundos. | |  | |  | |  |  |  | |

## *Módulo de Registro:*

***Introducción***: Para este Modulo de Registro se intenta demostrar a través de pruebas de Sistema que después de realizar el registro de un usuario se capturan los datos proporcionados por este.

***Propósito***: Asegurar que la validación y generación de los datos es correcta, con su relación correspondiente dentro del sistema.

### Prueba Caja negra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Logotipo  Descripción generada automáticamente | ***Módulo de Inicio de Sesión*** | ***Versión 2*** | ***Tester***: Morales Gonzales Juan Pablo | **Fecha de Prueba: 23/11/2024** |

|  |  |  |
| --- | --- | --- |
| Datos de Entrada | Clases validas | Clases no validas |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Casos de Prueba | Clases validas | Clases no validas | Salida |
|  |  |  |  |
|  |  |  |  |

### Prueba de Fiabilidad

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prueba de Fiabilidad | | | | | | | | | | |
| Logotipo  Descripción generada automáticamente | | ***Módulo de Inicio de Sesión*** | | ***Versión 1*** | | ***Tester***: Medellín Redondo Jorge Daniel | | | | **Fecha de Prueba: 23/11/2024** |
| ***Resultado*** ***esperado***: El formulario debe procesarse en menos de 2 segundos. Cualquier error (como campos vacíos o datos incorrectos) debe mostrar un mensaje claro y comprensible. El módulo de registro debe estar operativo en al menos el 99.9% del tiempo. Los datos del usuario deben guardarse sin alteraciones y sin pérdida. | | | | | | | | | | |
| ***Tópico para Evaluar*** | ***Argumento*** | | ***Bueno*** | | ***Medio*** | | ***Aceptable*** | ***Malo*** | ***Observaciones*** | |
| Tiempo de respuesta | El sistema debe responder rápidamente al enviar el formulario de registro. | |  | |  | |  |  |  | |
| Gestión de errores | |  | | --- | | El sistema debe manejar correctamente los errores en caso de fallos en el registro. |  |  | | --- | |  | | |  | |  | |  |  |  | |
| Disponibilidad | El sistema debe estar disponible y funcionando en todo momento. | |  | |  | |  |  |  | |
| Integridad de datos | Los datos ingresados deben guardarse correctamente en el sistema. | |  | |  | |  |  |  | |
| Conexión al servidor | El sistema debe conectarse al servidor sin interrupciones ni errores inesperados. | |  | |  | |  |  |  | |
| Resiliencia | El sistema debe continuar funcionando incluso si ocurre un error menor (como una desconexión breve). | |  | |  | |  |  |  | |

### Mantenibilidad

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Prueba de Fiabilidad | | | | | | | | | | |
| Logotipo  Descripción generada automáticamente | | ***Módulo de Inicio de Sesión*** | | ***Versión 1*** | | ***Tester***: Guerra Ruiz Daniel | | | | **Fecha de Prueba: 24/11/2024** |
| ***Resultado*** ***esperado***: El código debe ser fácil de leer y comprender, con comentarios que expliquen funciones o secciones importantes. Los errores deben ser identificables con herramientas de desarrollo estándar, y los mensajes de error deben ser informativos. El tiempo promedio para realizar ajustes debe ser acorde con las mejores prácticas de desarrollo. | | | | | | | | | | |
| ***Tópico para Evaluar*** | ***Argumento*** | | ***Bueno*** | | ***Medio*** | | ***Aceptable*** | ***Malo*** | ***Observaciones*** | |
| Claridad del código | El código debe estar bien estructurado, con comentarios claros que expliquen las funciones clave. | |  | |  | |  |  |  | |
| Modularidad | |  | | --- | | Las funciones deben estar separadas en componentes reutilizables y bien definidos. |  |  | | --- | |  | | |  | |  | |  |  |  | |
| Facilidad de depuración | Los errores deben ser fáciles de identificar y corregir. | |  | |  | |  |  |  | |
| Estandarización | El código debe seguir estándares comunes (formato, convenciones de nombres). | |  | |  | |  |  |  | |
| Compatibilidad con versiones anteriores | El módulo debe poder adaptarse a futuras actualizaciones sin necesidad de reescrituras completas. | |  | |  | |  |  |  | |
| Tiempos de actualización | El tiempo necesario para realizar modificaciones o ajustes debe ser razonable. | |  | |  | |  |  |  | |

## *Módulo de casos Actuales:*

### Submódulo de “Visualización de todos los casos”

Prueba de caja blanca

### Submódulo de “Detalles de todos los casos”

Prueba de Usabilidad

### Submódulo de “Asignar nuevo caso”

Prueba de caja negra.