

**PLAN DE PRUEBAS (Sabalí)**

**Shop and Service**

BITÁCORA DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Modificado por | Versión | Descripción |
| 26/02/2025 | Juan David López Correa -  Tomas Alejandro Gómez Restrepo | 1.0 | Creación de la introducción, objetivo general y estrategia de pruebas |
| 01/03/2025 | Juan David López Correa -  Tomas Alejandro Gómez Restrepo | 2.0 | Propósito y documentación |
| 06/03/2025 | Juan David López Correa -  Tomas Alejandro Gómez Restrepo | 3.0 | Características y criterios |
| 06/03/2025 | Juan David López Correa -  Tomas Alejandro Gómez Restrepo | 4.0 | Finalización del plan |

FORMATO PRELIMINAR AL DOCUMENTO

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Título: | **PLAN DE PRUEBAS SISTEMA SABALÍ** | | | | |
| ID: | **PLAN\_001 - SABALI** | | | | |
| Fecha elaboración: | 26/02/2025 | | | | |
| Palabras Claves: | Plan de pruebas, backend, pruebas funcionales, validación de datos, Node.js, SQL Server, validación de seguridad. | | | | |
| Formato: | .docx | | Lenguaje: | | Español |
| Versión: | | 1.0 | | Estado: | Aprobado |
| Autor (es): | Juan David López Correa -  Tomas Alejandro Gómez Restrepo | | | Firmas: |  |
| Revisó: | Juan David López Correa -  Tomas Alejandro Gómez Restrepo | | | Tomas Gómez  Juan López |
| Aprobó: | Juan David López Correa -  Tomas Alejandro Gómez Restrepo | | |  |

TABLA DE CONTENIDO

[BITÁCORA DE CAMBIOS 2](#_Toc475111325)

[FORMATO PRELIMINAR AL DOCUMENTO 3](#_Toc475111326)

[TABLA DE CONTENIDO 4](#_Toc475111327)

[INDICE DE TABLAS 5](#_Toc475111328)

[1. INTRODUCCIÓN 6](#_Toc475111329)

[**1.1.** **OBJETIVO GENERAL** 6](#_Toc475111330)

[**1.2.** **ESTRATEGIA DE PRUEBAS** 6](#_Toc475111331)

[**1.3.** **ALCANCE** 6](#_Toc475111332)

[**1.4.** **PROPOSITO** 6](#_Toc475111333)

[2. ENTREGABLES 7](#_Toc475111334)

[**2.1.** **DOCUMENTACIÓN A ENTREGAR** 7](#_Toc475111335)

[3. CARACTERISTICAS A SER PROBADAS 8](#_Toc475111336)

[4. CARACTERISTICAS A NO SER PROBADAS 9](#_Toc475111337)

[5. CRITERIOS DE APROBACIÓN Y FALLO 10](#_Toc475111338)

[6. CRITERIOS DE SUSPENSIÓN Y REANUDACIÓN 11](#_Toc475111339)

[7. TAREAS DE LAS PRUEBAS 12](#_Toc475111340)

[8. NECESIDADES AMBIENTALES 13](#_Toc475111341)

[**8.1.** **HARDWARE** 13](#_Toc475111342)

[**8.2.** **PLANEACIÓN DE COSTOS** 13](#_Toc475111343)

[**8.3.** **SUT (SISTEMA BAJO PRUEBAS)** 14](#_Toc475111344)

[**8.4.** **TEST-WARE.** 14](#_Toc475111345)

[9. CAPACITACIONES 15](#_Toc475111346)

[10. RIESGOS 16](#_Toc475111347)

[11. LABORATORIO DE USABILIDAD 17](#_Toc475111348)

INDICE DE TABLAS

[Tabla 1: Documentación a entregar. 7](#_Toc475111349)

[Tabla 2: Características a ser probadas. 8](#_Toc475111350)

[Tabla 3: Características a no ser probadas. 9](#_Toc475111351)

[Tabla 4: Criterios de aprobación y fallo. 10](#_Toc475111352)

[Tabla 5: Criterios de suspensión y reanudación. 11](#_Toc475111353)

[Tabla 6: Tareas de las pruebas. 12](#_Toc475111354)

[Tabla 7: Necesidades ambientales de hardware. 13](#_Toc475111355)

[Tabla 8: Planeación de costos. 13](#_Toc475111356)

[Tabla 9: Sistema bajo pruebas. 14](#_Toc475111357)

[Tabla 10: Test-ware. 14](#_Toc475111358)

[Tabla 11: Capacitaciones. 15](#_Toc475111359)

[Tabla 12: Riesgos. 16](#_Toc475111360)

[Tabla 13: Laboratorio de usabilidad. 17](#_Toc475111361)

1. INTRODUCCIÓN

Este documento establece un plan detallado de pruebas para la validación del backend del sistema SABALÍ, desarrollado en **TypeScript y Node.js** con una base de datos en **SQL Server**. Este sistema está diseñado para gestionar la información de una empresa de tercerización de servicios textiles, eliminando el uso de registros en papel y archivos Excel, garantizando la seguridad y disponibilidad de los datos. La implementación de un sistema centralizado permitirá a la empresa administrar clientes, insumos y proveedores de manera eficiente, con acceso seguro y controlado.

Este plan de pruebas se centra en validar que cada funcionalidad del sistema cumpla con los requisitos del negocio, garantizando integridad de datos, robustez y seguridad en cada operación. Se implementarán diversas estrategias de pruebas, incluyendo **pruebas unitarias, pruebas funcionales y pruebas de regresión**, empleando herramientas especializadas como **Postman, Jest, Supertest, Karate Framework, Gherkin y Cucumber**. La ejecución de pruebas estructuradas asegurará que el sistema esté libre de fallos antes de su despliegue en producción.

* 1. **OBJETIVO GENERAL**

El propósito de este plan es validar la correcta implementación y funcionamiento del backend del sistema SABALÍ, asegurando que cada módulo y servicio expuesto funcione conforme a los requisitos establecidos. Se verificará el cumplimiento de los criterios de seguridad, disponibilidad y eficiencia, previniendo vulnerabilidades y garantizando una óptima experiencia para los usuarios.

* 1. **ESTRATEGIA DE PRUEBAS**
* El enfoque de pruebas contempla el uso de metodologías automatizadas y manuales para garantizar la cobertura total del sistema. Se ejecutarán pruebas en entornos controlados, replicando condiciones reales de uso para evaluar la estabilidad, seguridad y escalabilidad de la plataforma.
* Se implementarán **pruebas automatizadas,** asegurando que el sistema cumpla con lo requisitos del sistema, por otro lado, se realizarán **pruebas de regresión** periódicamente bajo scripts automatizados para verificar que las actualizaciones no introduzcan errores en funcionalidades previas.
* Para la validación de la base de datos, se ejecutarán **pruebas de integridad de datos**, asegurando que cada operación CRUD se refleje correctamente sin pérdida de información ni inconsistencias en la base de datos SQL Server.
* Como parte del enfoque DevSecOps, se implementarán pruebas de seguridad automatizadas y manuales en cada iteración del desarrollo. Se ejecutarán análisis estático de código (SAST) para identificar vulnerabilidades en el backend y pruebas dinámicas de seguridad (DAST) para evaluar la seguridad de los endpoints de la API.
* También se ejecutarán análisis de seguridad en las dependencias del backend para detectar paquetes vulnerables en Node.js, empleando herramientas como Snyk o Dependabot.
  1. **ALCANCE**

El presente plan de pruebas abarca la validación completa del backend del sistema SABALÍ, incluyendo todas las funcionalidades asociadas a la gestión de usuarios, clientes, insumos y proveedores. Se realizarán pruebas exhaustivas sobre la autenticación de usuarios, garantizando que cada usuario tenga acceso solo a la información que le corresponde. También se verificará el correcto funcionamiento de las operaciones CRUD sobre cada módulo del sistema, asegurando que los datos sean almacenados, modificados y eliminados de manera adecuada en la base de datos. Adicionalmente, se validará la integridad y consistencia de la información, así como el correcto manejo de errores y la capacidad del sistema para soportar múltiples solicitudes simultáneas sin afectar el rendimiento. Se excluyen del alcance de este plan las pruebas relacionadas con la interfaz gráfica del sistema y cualquier integración con sistemas externos, ya que estos aspectos serán evaluados en un plan de pruebas separado.

* 1. **PROPOSITO**

Este plan de pruebas tiene como finalidad garantizar que el backend del sistema SABALÍ sea evaluado bajo múltiples escenarios de prueba para identificar y mitigar posibles fallos antes de su implementación en producción. Se busca proporcionar un marco estructurado que permita asegurar que todas las funcionalidades sean evaluadas a profundidad, previniendo errores en la manipulación de datos, autenticación, rendimiento y seguridad del sistema.

1. ENTREGABLES
   1. **DOCUMENTACIÓN A ENTREGAR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DOCUMENTO** | **PERSONA QUIEN ENTREGA** | **PERSONA QUIEN RECIBE** | **FECHA PLANEADA** | **FECHA DE ENTREGA** |
| Casos de prueba | Analista QA | Product Owner | 26/02/2025 | 13/03/2025 |
| Especificaciones del diseño de pruebas | Analista QA | Product Owner | 26/02/2025 | 13/03/2025 |
| Reportes de errores y defectos detectados | Analista QA | Líder técnico | 26/02/2025 | 27/03/2025 |
| Evidencias de pruebas realizadas | Analista QA | Product Owner | 26/02/2025 | 27/03/2025 |
| Documentación técnica | Desarrollador | Líder técnico | 26/02/2025 | 27/03/2025 |

Tabla 1: Documentación a entregar.

1. CARACTERISTICAS A SER PROBADAS

Las pruebas se enfocarán en la validación de las siguientes características:

|  |  |  |
| --- | --- | --- |
| **CARACTERISTICA** | **DESCRIPCIÓN** | **MODULO** |
| Autenticación | Autenticación de usuario | Log in |
| Leer | Obtener clientes en base de datos | Clientes |
| Crear | Crear clientes | Clientes |
| Actualizar | Actualización datos de cliente | Clientes |
| Eliminar | Eliminar usuarios de base de datos | Clientes |
| Leer | Obtener usuarios en base de datos | Usuarios |
| Crear | Crear usuarios | Usuarios |
| Actualizar | Actualización datos de usuarios | Usuarios |
| Eliminar | Eliminar usuarios de base de datos | Usuarios |
| Leer | Obtener proveedores en base de datos | Proveedores |
| Crear | Crear proveedores | Proveedores |
| Actualizar | Actualización datos de proveedores | Proveedores |
| Eliminar | Eliminar proveedores de base de datos | Proveedores |
| Leer | Obtener insumos en base de datos | Insumos |
| Crear | Crear insumo | Insumos |
| Actualizar | Actualización datos de insumos | Insumos |
| Eliminar | Eliminar insumos de base de datos | Insumos |
| Integridad de datos | Consistencia en base de datos | Base de datos |
| Seguridad | Vulnerabilidad del sistema | Backend |

Tabla 2: Características a ser probadas.

1. CARACTERISTICAS A NO SER PROBADAS

Dentro del presente plan de pruebas no se incluirán las siguientes validaciones:

|  |  |  |  |
| --- | --- | --- | --- |
| **CARACTERISTICA** | **DESCRIPCIÓN** | **JUSTIFICACIÓN** | **RIESGO** |
| Interfaz grafica | Front end | No fue añadida a la primera entrega del proyecto | Pruebas e2e de la aplicación |
| Integración con sistemas externos | Apis externas | Cualquier tipo de api de autorización | Brechas de seguridad en Loguin |
| Pruebas móviles | Pruebas en dispositivos móviles | En la primera entrega no se tendrán aplicaciones | Ninguno visto |

Tabla 3: Características a no ser probadas.

1. CRITERIOS DE APROBACIÓN Y FALLO

El sistema será aprobado si:

* Se ejecutan el **95% de las pruebas con éxito**.
* Se garantiza un tiempo de respuesta **inferior a 2 segundos** en el **95% de las solicitudes**.
* No se detectan **vulnerabilidades críticas** de seguridad.

El sistema fallará si:

* Se identifican errores en la autenticación de usuarios.
* Existen fallas en la integridad de los datos almacenados.
* Se detectan vulnerabilidades que comprometan la seguridad de la información.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID CRITERIO** | **DESCRIPCIÓN** | **APROBACIÓN** | **FALLO** |
|  |  |  |  |
|  |  |  |  |

Tabla 4: Criterios de aprobación y fallo.

1. CRITERIOS DE SUSPENSIÓN Y REANUDACIÓN

Las pruebas podrán suspenderse si se presentan fallos críticos en el sistema que impidan la ejecución normal de las pruebas, como caídas en la base de datos, errores graves en la autenticación de apis o tiempos de respuesta excesivamente altos. Las pruebas se reanudarán una vez que los errores críticos hayan sido corregidos y se realicen validaciones previas para garantizar la estabilidad del sistema antes de continuar con las pruebas.

|  |  |
| --- | --- |
| **CRITERIO DE SUSPENSIÓN** | **CRITERIO DE REANUDACIÓN** |
| Fallas críticas en la base de datos que impidan la ejecución de las pruebas. | Corrección de errores en la base de datos y validación de su estabilidad. |
| Errores graves en la autenticación y autorización de apis. | Implementación de correcciones y validación de acceso adecuado. |

Tabla 5: Criterios de suspensión y reanudación.

1. TAREAS DE LAS PRUEBAS

Las pruebas se ejecutarán en fases bien definidas, asegurando una validación completa del sistema:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TAREA** | **DESCRIPCIÓN** | **FECHA INICIO** | **FECHA FIN** | **RESPONSABLE** | **ROL** |
| Definición | Definición de casos de prueba. | Semana 1 | Semana 2 | Analista QA | QA |
| Configuración | Configuración de entornos. | Semana 2 | Semana 3 | Analista QA | QA |
| Ejecución | Ejecución de pruebas manuales. | Semana 2 | Semana 3 | Analista QA | QA |
| Validación | Validación de pruebas funcionales. | Semana 3 | Semana 4 | Analista QA | QA |
| Análisis | Ejecutar análisis de código (SAST) y evaluación de dependencias para detectar vulnerabilidades internas. | Semana 3 | Semana 4 | Analista QA | QA |
| Realización | Realizar pruebas dinámicas en los endpoints de la API para detectar vulnerabilidades externas. | Semana 4 | Semana 5 | Analista QA | QA |
| Evaluación | Evaluación de pruebas de carga y seguridad. | Semana 4 | Semana 5 | Analista QA | QA |
| Generación | Generación de reportes y resolución de fallos detectados. | Semana 4 | Semana 5 | Analista QA | QA |
| Automatización | Automatización de casos. | Semana 5 | Semana 6 | Analista QA | QA |
| Regresión | Validación de pruebas de regresión. | Semana 6 | Semana 7 | Analista QA | QA |

Tabla 6: Tareas de las pruebas.

1. NECESIDADES AMBIENTALES
   1. **HARDWARE**

|  |  |  |  |
| --- | --- | --- | --- |
| **DISPOSITIVO** | **MARCA** | **CARACTERISTICAS** | **¿TENEMOS EL EQUIPO?** |
| Servidor de pruebas | Dispuesto por el equipó | 16 GB RAM, procesador 8 núcleos | SI |

Tabla 7: Necesidades ambientales de hardware.

* 1. **PLANEACIÓN DE COSTOS**

El proceso de pruebas contempla herramientas de automatización y servidores de prueba. La implementación de herramientas como Postman y Karate framework, Gherkin, cucumber, y el mantenimiento de la infraestructura tecnológica necesaria para ejecutar las pruebas en entornos aislados.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID RECURSO** | **TIPO DE RECURSO** | **FORMA DE ADQUISICIÓN** | **UNIDAD** | **COSTO DE RESURSO X UNIDAD** | **HORAS** | **CANTIDAD** | **TOTAL** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Tabla 8: Planeación de costos.

* 1. **SUT (SISTEMA BAJO PRUEBAS)**

El sistema bajo pruebas es el backend de la aplicación SABALI, el cual incluye la gestión de usuarios, clientes, insumos y proveedores. Se evaluarán las interacciones con la base de datos, la correcta implementación de las API REST y la seguridad de los datos almacenados y transmitidos.

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULO** | **¿DEPENDENCIA ENTRE MODULOS?** | | **REQUERIMIENTOS** |
| Gestión de usuarios | No |  | CRUD apis |
| Gestión de insumos | Si | Con Proveedor | Validación de stock y manejo de proveedores. |
| Gestión de proveedores | No |  | CRUD completo |
| Gestión de clientes | No |  | Persistencia en base de datos y API funcional. |

Tabla 9: Sistema bajo pruebas.

* 1. **TEST-WARE.**

El test-ware incluirá la documentación de los casos de prueba, scripts de pruebas automatizadas, configuraciones del entorno de pruebas, reportes de errores y documentación técnica del proceso de validación.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ADMINITRACION DE LA CONFIGURACIÓN** | **USUARIOS** | **ACCESOS A BD** | **GUIAS DE PRUEBAS** | **CASOS DE PRUEBAS** | **DOCUMENTACION ESPECIFICA DEL PROYECTO** |
| Configuración de entorno de pruebas | Testers y desarrolladores | Roles de acceso a SQL Server | Manual de ejecución de pruebas | Casos de prueba documentados | Documentación técnica sobre pruebas |
|  |  |  |  |  |  |

Tabla 10: Test-ware.

1. CAPACITACIONES

No se realizarán capacitaciones, ya que el sistema es un backend y no está dirigido a usuarios finales. Todas las pruebas serán ejecutadas por el equipo de calidad y desarrollo, quienes ya cuentan con el conocimiento necesario sobre el funcionamiento del sistema.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **INSTRUCTOR** | **PERSONA A CAPACITAR** | **CAPACITACIÓN** | **FECHA INICIO** | **FECHA FIN** | **DURACIÓN**  **HRS** | **COSTO** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Tabla 11: Capacitaciones.

1. RIESGOS

Los principales riesgos identificados en el proceso de pruebas incluyen posibles fallos en la base de datos que afecten la integridad de la información, vulnerabilidades de seguridad que comprometan la protección de los datos y tiempos de respuesta elevados en condiciones de alta carga. Se han implementado estrategias de mitigación, como la ejecución de pruebas de estrés y auditorías de seguridad periódicas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id Riesgo |  |  |  |  |
| Nombre |  |  |  |  |
| Descripción del riesgo |  |  |  |  |
| Estado inicial |  |  |  |  |
| Consecuencias |  |  |  |  |
| Probabilidad de ocurrencia |  |  |  |  |
| Impacto |  |  |  |  |
| Prioridad |  |  |  |  |
| Clasificación |  |  |  |  |
| Síntomas |  |  |  |  |
| Tolerancias |  |  |  |  |
| Acciones preventivas |  |  |  |  |
| Acciones correctivas |  |  |  |  |

Tabla 12: Riesgos.

1. LABORATORIO DE USABILIDAD

Dado que el sistema es un backend, no se realizarán pruebas de usabilidad para usuarios finales. Sin embargo, se validará la eficiencia de los endpoints en términos de facilidad de uso para desarrolladores y la correcta implementación de respuestas claras y documentadas en la API REST.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CITA** | **CLASIFICACIÓN** | **DESCRIPCIÓN** | **DURACIÓN**  **HRS** | **FECHA** | **USUARIOS** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Tabla 13: Laboratorio de usabilidad.