Documento de Pruebas

uan Tobón, Jhon Sandoval, Andrés Carrasquilla, Jhon Matthew, Yohan Piñarte, Diego Aguilera,
Universidad de los Llanos, Colombia

1. Introducción

1.1 Objetivo

El presente documento tiene como objetivo principal establecer un plan detallado para la realización de pruebas del sistema desarrollado para la gestión, visualización y análisis de resultados de las pruebas Saber Pro. Estas pruebas buscan asegurar que el software cumpla con los requisitos funcionales y no funcionales especificados, garantizando su correcto funcionamiento, fiabilidad, seguridad y rendimiento óptimo.

Este documento está dirigido a todos los miembros del equipo de desarrollo, aseguramiento de calidad (QA), así como a los interesados clave del proyecto, incluyendo administradores y usuarios finales. A través de este documento, se pretende facilitar la identificación temprana de errores o fallos, contribuir a la mejora continua del sistema y validar que las funcionalidades implementadas cumplen con las expectativas y necesidades del usuario.

En particular, se cubrirán pruebas enfocadas en las diferentes capas del sistema: la interfaz web para el usuario, el backend construido con Spring Boot que expone servicios REST, y la base de datos PostgreSQL que almacena y administra la información. El objetivo es también evaluar la integración correcta entre estos componentes y garantizar que la plataforma ofrezca una experiencia robusta, segura y eficiente a los usuarios.

1.2 Alcance

El alcance de las pruebas comprende todas las funcionalidades críticas y complementarias del sistema Saber Pro. Esto incluye, pero no se limita a:

- **Autenticación y autorización:** Verificación del correcto funcionamiento del inicio de sesión, control de acceso basado en roles (administrador, docente, estudiante) y seguridad en la gestión de sesiones mediante tokens JWT.

- **Carga y validación de datos:** Pruebas de la funcionalidad que permite al administrador subir archivos CSV con los resultados de las pruebas Saber, asegurando que el sistema valide la estructura, formato y consistencia de los datos antes de almacenarlos.

- **Consulta y filtrado de resultados:** Evaluación de la capacidad del sistema para realizar búsquedas dinámicas y aplicadas sobre la base de datos, utilizando múltiples filtros como Documento, programa académico y área.

- **Visualización de datos:** Validación de la correcta generación y despliegue de gráficos y tablas que representan el desempeño de los estudiantes, facilitando el análisis visual de

los resultados.

- **Generación y exportación de reportes:** Comprobación de la funcionalidad que permite a los usuarios descargar reportes en formatos estándar (Excel), garantizando que los datos exportados sean precisos y estén bien formateados.

- **Seguridad y rendimiento:** Pruebas relacionadas con la protección de datos, resistencia a accesos no autorizados, y tiempo de respuesta adecuado del sistema bajo diferentes cargas.

1.3 Propósito del Documento de Pruebas

Este documento busca formalizar el proceso de pruebas que permitirá validar el sistema de manera sistemática y documentada. Sirve como guía para el equipo de QA y como referencia para el equipo de desarrollo y la gerencia de proyecto, asegurando que:

- Se entiendan claramente las funcionalidades que serán probadas.

- Se definen los criterios para considerar las pruebas exitosas o fallidas.

- Se planifican recursos, tiempos y responsabilidades.

- Se facilita la trazabilidad entre requisitos, casos de prueba y resultados.

Además, el documento contribuye a la mejora continua del producto al permitir el seguimiento y análisis de defectos, así como la identificación de oportunidades para optimizar procesos y calidad.

1.4 Referencias

Las siguientes fuentes y estándares han sido consultados para la elaboración del documento de pruebas y el desarrollo del sistema:

- IEEE Std 829-2008, “Standard for Software and System Test Documentation.”

- Documentación oficial de Spring Boot: https://spring.io/projects/spring-boot

- Documentación oficial PostgreSQL: https://www.postgresql.org/docs/

- Manuales y lineamientos ICFES sobre pruebas Saber: https://www.icfes.gov.co

- Documentación técnica sobre JWT y seguridad web.

2. Tipos de Pruebas

En esta sección se describen los diferentes tipos de pruebas que se realizan para garantizar la calidad y el correcto funcionamiento del sistema Saber Pro. Cada tipo de prueba tiene un propósito específico y se enfoca en diferentes aspectos del sistema, asegurando una cobertura integral de los requisitos.

2.1 Pruebas Unitarias

Las pruebas unitarias están orientadas a verificar el correcto funcionamiento de las unidades más pequeñas de código, tales como funciones, métodos o clases individuales dentro del backend desarrollado en Spring Boot. Estas pruebas se realizan de forma aislada para asegurar que cada componente cumple con su comportamiento esperado antes de integrarse al sistema completo.

- **Objetivo:** Detectar errores en la lógica interna del código, validar que los métodos devuelvan resultados correctos para distintos inputs, y asegurar la robustez ante casos límite o datos incorrectos.

- **Alcance:** Se cubrirán los servicios, controladores y repositorios que gestionan la lógica de negocio y acceso a datos.

2.2 Pruebas de Integración

Las pruebas de integración tienen como objetivo verificar la correcta interacción entre diferentes componentes del sistema, principalmente entre el backend Spring Boot y la base de datos PostgreSQL, así como entre el backend y la interfaz a través de servicios REST.

- **Objetivo:** Confirmar que las consultas, inserciones y actualizaciones en la base de datos funcionan correctamente cuando son invocadas desde el backend, y que las respuestas de la API REST cumplen con el formato y contenido esperado.

- **Herramientas:** Spring Test, Postman para llamadas a API.

- **Alcance:** Pruebas de endpoints REST, validación de respuestas JSON, manejo de errores y autenticación.

2.3 Pruebas Funcionales

Estas pruebas verifican que el sistema cumpla con los requisitos funcionales especificados en el documento de requisitos. Se realizan desde la perspectiva del usuario, comprobando que cada función del sistema opere correctamente en un entorno real.

- **Objetivo:** Asegurar que las funcionalidades como inicio de sesión, carga y validación de archivos, aplicación de filtros, visualización de resultados, y generación de reportes respondan a las expectativas.

- **Alcance:** Interfaces de usuario, operaciones de backend accesibles desde la UI, generación y exportación de reportes.

2.4 Pruebas de Rendimiento

Se realizarán pruebas orientadas a medir y evaluar el tiempo de respuesta del sistema y su comportamiento bajo diferentes niveles de carga, con el fin de garantizar que el sistema sea eficiente y escalable.

- **Objetivo:** Validar que el tiempo de respuesta ante solicitudes comunes (como consultas y filtros) sea inferior a 2 segundos, y que el sistema mantenga estabilidad y desempeño

adecuado ante múltiples usuarios simultáneos.

- **Alcance:** Pruebas sobre la API REST, consultas a la base de datos, y generación de reportes bajo carga.

2.5 Pruebas de Seguridad

Estas pruebas buscan garantizar la protección de los datos y el acceso restringido sólo a usuarios autorizados, así como la resistencia ante ataques comunes como inyección SQL, acceso sin autenticación o manipulación de tokens.

- **Objetivo:** Validar la implementación segura de la autenticación basada en JWT, el control de permisos por roles, la protección de endpoints sensibles, y la seguridad en la gestión de sesiones.

- **Herramientas:** Herramientas de escaneo de vulnerabilidades y pruebas manuales con Postman.

- **Alcance:** Todos los servicios REST, mecanismos de autenticación y autorización.

2.6 Pruebas de Usabilidad (Opcional)

Aunque no es el foco principal, se realizarán pruebas básicas de usabilidad para asegurar que la interfaz sea intuitiva y amigable para los usuarios finales, facilitando la navegación y comprensión de los datos.

- **Objetivo:** Identificar posibles mejoras en la experiencia de usuario y en la accesibilidad de la plataforma.

- **Herramientas:** Revisiones manuales y feedback de usuarios piloto.

3. Plan de Pruebas

En esta sección se presenta el plan detallado para la ejecución de las pruebas del sistema Saber Pro. Se definen las estrategias, actividades, recursos y criterios que guiarán el proceso para asegurar una verificación rigurosa y sistemática del software.

3.1 Estrategia de Pruebas

La estrategia de pruebas se basa en un enfoque escalonado que abarca desde pruebas unitarias hasta pruebas de integración, funcionales, de rendimiento y seguridad, asegurando una cobertura completa de los aspectos técnicos y funcionales del sistema. Se combinarán pruebas automatizadas y manuales para optimizar recursos y tiempo.

- **Pruebas Unitarias:** Implementadas y ejecutadas por el equipo de desarrollo durante el ciclo de desarrollo usando frameworks como JUnit y Mockito.

- **Pruebas de Integración:** Realizadas mediante pruebas automáticas y con herramientas de API testing (Postman) para validar la interacción entre componentes.

- **Pruebas Funcionales:** Ejecutadas manualmente por el equipo de QA y con apoyo de automatización en Selenium para validar los flujos principales del sistema.

- **Pruebas de Rendimiento:** Realizadas con herramientas especializadas para simular cargas y medir tiempos de respuesta.

- **Pruebas de Seguridad:** Incluyen escaneo de vulnerabilidades y pruebas manuales de autenticación y autorización.

3.2 Actividades y Cronograma

El proceso de pruebas se divide en fases con actividades claramente definidas para facilitar el seguimiento y control:

Fase	**Actividades Principales**	**Responsable**	
Preparación	Definición de casos de prueba y entornos	Equipo de QA	1 semana
Pruebas Unitarias	Desarrollo y ejecución	Equipo de desarrollo	Continuo durante el desarrollo
Pruebas de Integración	Ejecución de pruebas API y base de datos	QA y desarrollo	1 semana
Pruebas Funcionales	Ejecución manual y automatizada de casos	Equipo de QA	2 semanas
Pruebas de Rendimiento	Simulación de carga y análisis	Equipo de QA	1 semana
Pruebas de Seguridad	Evaluación de autenticación y control de acceso	Equipo de QA	1 semana
Revisión y Reporte	Análisis de resultados y reporte final	Todos	1 semana

3.3 Recursos

- **Humanos:** Equipo de desarrollo (3 personas), equipo de QA (2 personas).

- **Herramientas:** IDEs para desarrollo y pruebas unitarias (IntelliJ), Postman, Selenium, JMeter, herramientas de escaneo de seguridad.

- **Infraestructura:** Servidor para despliegue backend con Java y PostgreSQL, entornos de pruebas configurados para simular condiciones reales.

3.4 Criterios de Entrada

- Disponibilidad del código funcional y estable en el repositorio.

- Entorno de pruebas configurado y accesible.
- Casos de prueba definidos y revisados.
- Datos de prueba preparados y validados.

3.5 Criterios de Salida

- Ejecución satisfactoria de todos los casos de prueba críticos.
- Identificación, reporte y corrección de defectos detectados.
- Validación del cumplimiento de requisitos funcionales y no funcionales.
- Aprobación formal del sistema para despliegue en producción.

3.6 Gestión de Defectos

Todos los defectos encontrados durante las pruebas serán registrados en un sistema de seguimiento (por ejemplo, Jira o Trello), clasificados según su gravedad y prioridad, asignados al equipo de desarrollo para su resolución y verificados posteriormente por QA.

4. Casos de Prueba

En esta sección se describen de manera detallada los casos de prueba diseñados para validar las funcionalidades del sistema Saber Pro. Cada caso de prueba especifica el objetivo, los datos de entrada, los pasos a seguir, los resultados esperados y los criterios de aceptación, con el fin de asegurar una ejecución rigurosa y reproducible durante las fases de prueba.

4.1 Caso de Prueba: Inicio de Sesión

- **Identificador:** CP001
- **Objetivo:** Verificar que el sistema permita el inicio de sesión correcto para usuarios con roles de Decano, Coordinador saber pro, Director de programa, comité de programa y profesor
- **Precondiciones:**
    - El usuario debe estar registrado en el sistema.
    - La plataforma debe estar disponible y el servidor activo.
- **Datos de Entrada:**
    - Usuario: correo electrónico o nombre de usuario válido.
    - Contraseña: contraseña correcta asociada al usuario.

- **Procedimiento:**
    - Ingresar a la página de inicio de sesión.
    - Introducir credenciales válidas (usuario y contraseña).
    - Presionar el botón "Iniciar Sesión".
- **Resultado Esperado:**
    - El sistema autentica al usuario correctamente.
    - El usuario es redirigido a la página principal y muestra según su rol
    - Se muestra un mensaje de bienvenida personalizado.
- **Criterios de Aceptación:**
    - El sistema muestra una pantalla de inicio de sesión con campos para correo y contraseña.
    - Si el correo o la contraseña son incorrectos, se muestra un mensaje de error claro.
    - Si las credenciales son válidas, el usuario accede al menú principal del sistema.
    - El sistema reconoce el rol del usuario y muestra solo las funcionalidades permitidas.

4.2 Caso de Prueba: Carga de Archivo CSV con Resultados

- **Identificador:** CP002
- **Objetivo:** Validar que los roles autorizados puedan cargar archivos CSV con resultados de pruebas correctamente formateados.
- **Precondiciones:**
    - Usuario autenticado con rol de decano, coordinador saber pro o director de programa
    - Archivo CSV con formato válido y datos consistentes.
    - Datos de año y ciclo ingresados con anterioridad.
- **Datos de Entrada:**
    - Archivo CSV con columnas y datos conforme a la estructura esperada.
- **Procedimiento:**
    - Navegar a la sección de carga de archivos.
    - Seleccionar y cargar el archivo CSV.

    - Confirmar la carga y esperar la validación automática.

- **Resultado Esperado:**
    - El sistema valida la estructura y contenido del archivo.
    - Se almacenan los datos correctamente en la base de datos.
    - Se muestra un mensaje de confirmación exitosa.
- **Criterios de Aceptación:**

    - Si se intenta cargar el reporte sin los datos de año y ciclo, se muestra error.
    - Si el archivo es inválido o hay errores de estructura, el sistema notifica el problema y no permite la carga.
    - Los datos del reporte cargado se reflejan en el sistema y están disponibles para consulta

4.3 Caso de Prueba: Aplicación de Filtros Dinámicos

- **Identificador:** CP003
- **Objetivo:** Verificar que los usuarios puedan aplicar filtros para consultar resultados según diferentes criterios (cédula del estudiante, programa académico y área).
- **Precondiciones:**
    - Datos cargados y disponibles en el sistema.
- **Datos de Entrada:**
    - Selección de filtros específicos en la interfaz
- **Procedimiento:**
    - Acceder a la sección de visualización de resultados.
    - Seleccionar filtros deseados.
    - Aplicar filtros y esperar la carga de resultados.
- **Resultado Esperado:**
    - La lista se actualiza para reflejar los criterios seleccionados.
    - Los datos mostrados corresponden exclusivamente a los filtros aplicados.

- **Criterios de Aceptación:**
    - Respuesta rápida y sin errores.
    - Filtros combinables y funcionales en cualquier orden.

4.4 Caso de Prueba: Generación y Exportación de Reportes

- **Identificador:** CP004
- **Objetivo:** Asegurar que los usuarios autorizados puedan generar reportes basados en los resultados filtrados y exportarlos en formato Excel.
- **Precondiciones:**
    - Usuario autenticado con permisos para generar reportes.
    - Resultados visibles tras aplicar filtros.
- **Datos de Entrada:**
    - Selección de filtros y formato de exportación.
- **Procedimiento:**
    - Aplicar los filtros deseados.
    - Seleccionar la opción para generar reporte.
    - Descargar el reporte generado.
- **Resultado Esperado:**
    - El reporte refleja correctamente los datos filtrados.
    - El archivo exportado es legible y contiene todos los elementos esperados.
- **Criterios de Aceptación:**
    - Reporte generado sin errores ni datos faltantes.

5. Resultados y Seguimiento de Pruebas

Esta sección presenta el registro y análisis de los resultados obtenidos durante la ejecución de los casos de prueba, así como el seguimiento de los defectos encontrados para asegurar la calidad y estabilidad del sistema Saber Pro.

5.1 Registro de Resultados

Durante la fase de pruebas, cada caso de prueba se documenta cuidadosamente con el estado de su ejecución. Los posibles resultados registrados incluyen:

- Aprobado: El caso de prueba cumplió con los resultados esperados sin incidencias.
- Fallido: El caso de prueba no cumplió con los resultados esperados.
- Bloqueado: No fue posible ejecutar el caso de prueba debido a condiciones previas no cumplidas o problemas técnicos.

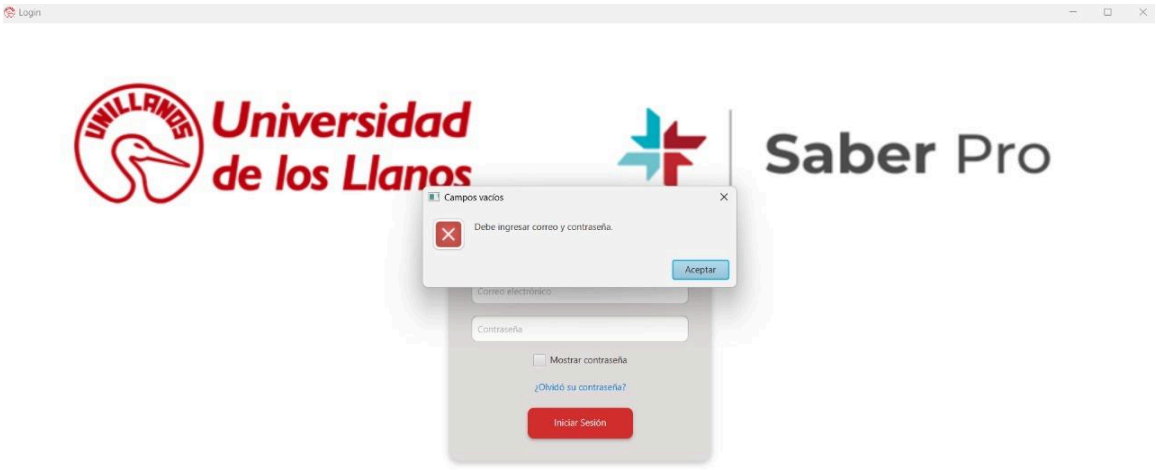
Se mantiene una tabla resumen donde se registran los resultados de cada caso de prueba para facilitar el análisis posterior.

Caso de Prueba	Estado	Observaciones	Fecha	Responsable
CP001	Aprobado	Inicio de sesión exitoso	2025-05-20	QA Equipo
CP002	Aprovado	Archivos cargados con éxito	2025-05-21	QA Equipo
CP003	Aprobado	Filtros funcionando correctamente	2025-05-22	QA Equipo
CP004	Aprobado	Reportes generados y exportados	2025-05-23	QA Equipo

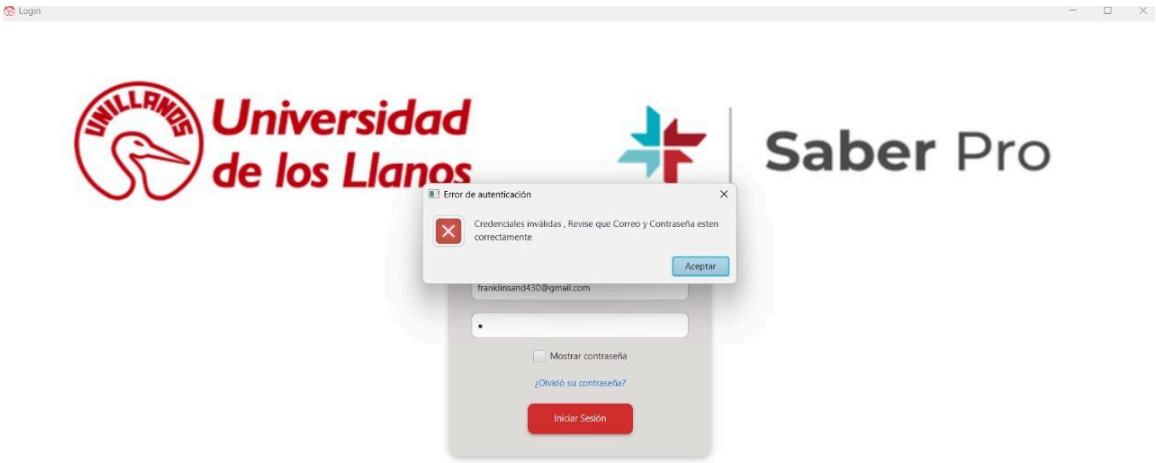
5.1.1 Evidencias Visuales de Fallos Detectados

Aquí se presentan capturas de pantalla que ilustran algunos errores relevantes encontrados durante las pruebas funcionales:

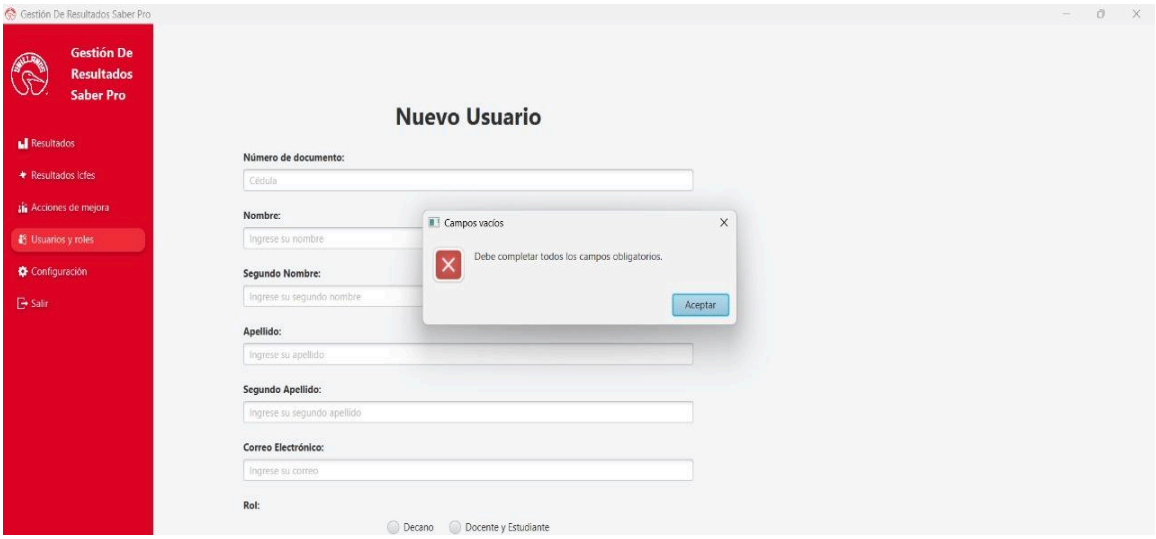
1. Figura filtros



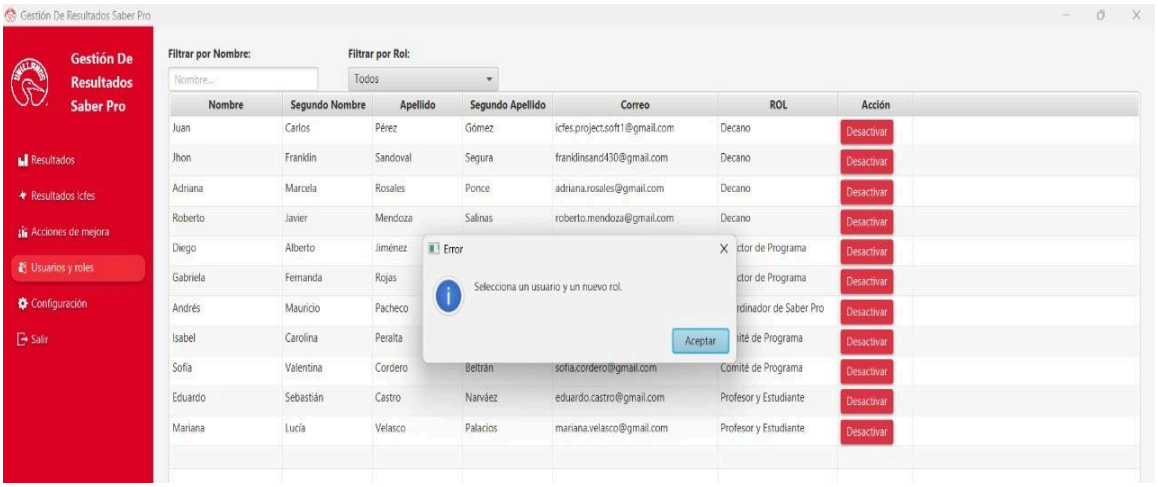
2. Figura Ingreso de credenciales vacío



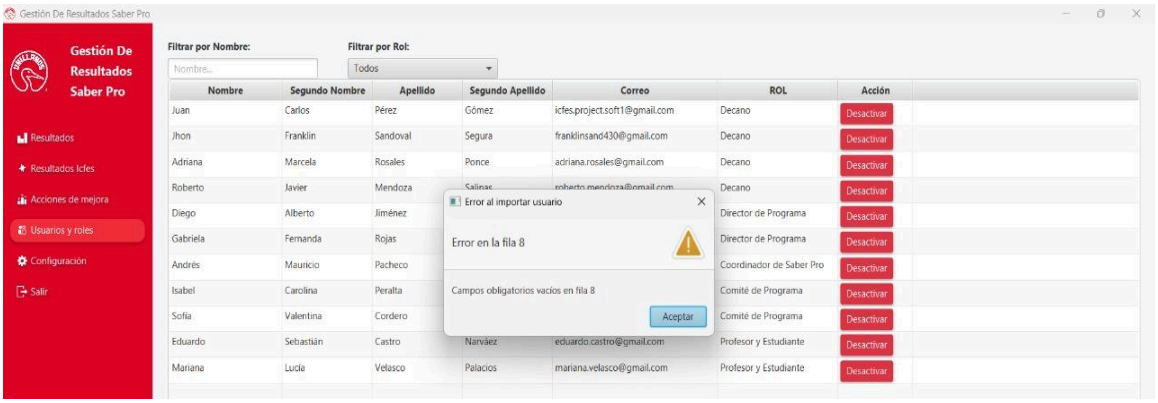
3. Figura Ingreso de credenciales erróneas



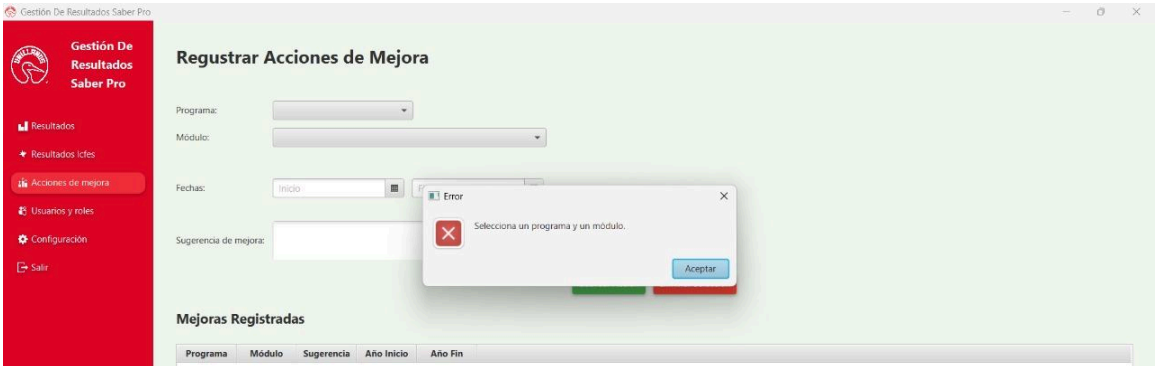
4. Figura Completar campos



5. Figura Elección de usuario y rol



6. Figura Campos Vacíos



7.Figura Selección de programa y módulo

5.2 Gestión de Defectos

Todos los defectos detectados durante las pruebas son reportados en la herramienta de gestión de incidencias.. Cada defecto incluye:

- **ID del defecto**
- **Descripción detallada**
- **Prioridad y severidad**

- **Estado (Abierto, En progreso, Resuelto, Cerrado)**
- **Persona asignada**
- **Fecha de reporte y resolución**

Los defectos se priorizan para su corrección, con especial atención a aquellos que afectan funcionalidades críticas como la autenticación o la carga de datos.

5.3 Análisis de Resultados

Al concluir la ejecución de los casos de prueba, se realiza un análisis para determinar:

- La tasa de éxito general de las pruebas.
- Los módulos con mayor número de fallos y posibles causas.
- El impacto de los defectos detectados en la operación del sistema.
- La efectividad de las correcciones aplicadas.

Este análisis permite definir acciones correctivas y mejorar el proceso de desarrollo y prueba.

5.4 Aprobación y Cierre de Pruebas

Una vez que todos los defectos críticos y de alta prioridad han sido corregidos y validados, y los casos de prueba críticos están aprobados, se realiza la aprobación formal del sistema para su despliegue.

El cierre del proceso de pruebas incluye la entrega de un informe final que resume los resultados, defectos encontrados y acciones realizadas.

6. Registro y Seguimiento de Defectos

El proceso de registro y seguimiento de defectos es fundamental para garantizar la calidad del sistema Saber Pro. Cada error detectado durante las pruebas es documentado exhaustivamente para facilitar su análisis y pronta resolución.

6.1 Documentación de Defectos

Para cada defecto encontrado se registra la siguiente información:

- **ID del Defecto:** Identificador único que facilita el seguimiento.
- **Descripción:** Detalle claro y conciso del problema detectado.
- **Pasos para Reproducir:** Secuencia exacta de acciones necesarias para replicar el error.
- **Gravedad:** Nivel de impacto del defecto en el sistema (Crítico, Alto, Medio, Bajo).

- **Estado:** Situación actual del defecto (Abierto, En Progreso, Resuelto, Cerrado).
- **Responsable:** Persona o equipo encargado de investigar y corregir el defecto.
- **Fecha de Reporte:** Momento en que se detectó y registró el defecto.
- **Fecha de Resolución:** Momento en que se corrigió y validó el defecto (cuando aplique).

6.2 Seguimiento y Revisión

El equipo de desarrollo y calidad realizará un seguimiento continuo de los defectos reportados, asegurando:

- Priorizar la resolución según la gravedad e impacto.
- Asignar responsables y fechas límite para la corrección.
- Validar las correcciones mediante la re-ejecución de los casos de prueba afectados.
- Actualizar el estado del defecto y documentar evidencias de la corrección.

Este proceso garantiza que ningún error crítico quede sin resolver antes del despliegue final del sistema.

7. Conclusiones

Tras la ejecución completa del plan de pruebas para el sistema Saber Pro, se resumen los hallazgos y aprendizajes obtenidos, proporcionando un análisis claro sobre la calidad actual del software y recomendaciones para futuros desarrollos.

7.1 Resumen de Resultados

Las pruebas funcionales y no funcionales realizadas confirmaron que la mayoría de las funcionalidades implementadas cumplen con los requisitos especificados, incluyendo:

- Inicio de sesión robusto y control de acceso por roles.
- Carga y validación adecuada de archivos CSV con resultados de pruebas.
- Aplicación eficiente de filtros dinámicos para consultas personalizadas.
- Generación y exportación correcta de reportes en formatos estándar.

7.2 Análisis de Defectos

Se identificaron algunos defectos principalmente relacionados con la validación de datos y manejo de archivos, los cuales fueron debidamente registrados y corregidos oportunamente. Los defectos restantes no críticos serán atendidos en futuras versiones para mejorar la estabilidad y

usabilidad del sistema.

7.3 Recomendaciones

Para fortalecer la calidad y mantener la continuidad del proyecto, se recomienda:

- Implementar pruebas automatizadas para reducir tiempos y errores humanos en ciclos futuros.
- Realizar sesiones periódicas de revisión de código y pruebas integrales tras cada actualización.
- Mejorar la documentación técnica para facilitar la incorporación de nuevos desarrolladores.
- Ampliar la cobertura de pruebas para incluir escenarios extremos y pruebas de carga.

Estas acciones permitirán que Saber Pro continúe evolucionando como una plataforma confiable y escalable, alineada con las necesidades de sus usuarios.