



Especificación de Requisitos de Software – Visualización de Resultados Saber Pro

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

Resumen - El proyecto Saber Pro consiste en crear una plataforma web para gestionar y visualizar los resultados de las pruebas Saber, cómo Saber 11 y Saber Pro. La plataforma debe incluir funcionalidades como inicio de sesión con control de acceso según el tipo de usuario (Decano, docente, estudiante), carga y validación de datos, filtros para análisis dinámico, visualización interactiva mediante gráficos y reportes, y exportación de datos en formatos como Excel o PDF.

I. INTRODUCCIÓN

El proyecto Saber Pro tiene como objetivo desarrollar una plataforma web que permita la gestión y visualización eficiente de los resultados de las pruebas Saber, incluyendo Saber 11 y Saber Pro. Estas pruebas son fundamentales para evaluar el desempeño académico de los estudiantes en Colombia, y contar con una herramienta digital que facilite el acceso, análisis y presentación de estos datos es vital para instituciones educativas, docentes y administradores.

La plataforma brindará funcionalidades integrales como autenticación y control de acceso, carga y validación de datos, análisis mediante filtros personalizados, generación de gráficos interactivos y exportación de informes en distintos formatos. Además, se implementará una arquitectura robusta basada en principios modernos como CQRS y Event-Driven, para garantizar escalabilidad, rendimiento y una experiencia de usuario óptima.

El desarrollo combinará tecnologías actuales para frontend, backend y bases de datos, asegurando una solución flexible y adaptable a las necesidades del sistema educativo.

II. MARCO TEÓRICO

El proyecto consiste en el desarrollo de una plataforma web destinada a la gestión, análisis y visualización de los resultados de las pruebas Saber 11 y Saber Pro. Estas pruebas estandarizadas, aplicadas en Colombia, tienen como finalidad evaluar el desempeño académico y profesional de los estudiantes en diferentes niveles educativos. La correcta administración y presentación de estos datos es fundamental para instituciones educativas, docentes y estudiantes, quienes requieren acceso eficiente y claro a esta información.

Para la construcción de la interfaz de usuario se eligió React, una biblioteca de JavaScript que facilita la creación de interfaces dinámicas y basadas en componentes reutilizables. Spring Boot permite una interacción fluida con la plataforma, posibilitando la aplicación de filtros, la visualización de gráficos interactivos y la generación de reportes, todo en un entorno responsivo y accesible desde distintos dispositivos.

En cuanto a la gestión de la información, se utilizó PostgreSQL como sistema de gestión de bases de datos relacional. PostgreSQL es conocido por su estabilidad y

capacidad para manejar datos estructurados de manera eficiente, lo cual es crucial para almacenar los resultados académicos y las diferentes entidades involucradas en el proyecto, garantizando integridad y seguridad en el manejo de la información.

El sistema está diseñado para permitir la autenticación de usuarios con diferentes roles, asegurando el acceso controlado a las funcionalidades según el perfil (Decano, Comité, docente, estudiante). Además, la plataforma incluye procesos para la carga y validación de datos, asegurando que la información ingresada cumpla con los formatos y criterios establecidos para mantener la calidad de los datos.

Para facilitar la comprensión, diseño y desarrollo del sistema, se utilizaron diagramas UML y diagramas de arquitectura. Estos modelos documentan la estructura del sistema, la interacción entre los distintos módulos (frontend, backend, base de datos), así como los flujos de datos y procesos internos, sirviendo como guía para el desarrollo y mantenimiento de la plataforma.

Finalmente, la plataforma permite aplicar filtros dinámicos para el análisis de los resultados, ofreciendo a los usuarios la capacidad de visualizar y exportar reportes en diferentes formatos, apoyando la toma de decisiones informadas basadas en los datos de las pruebas Saber.

III. OBJETIVOS

Diseñar e implementar una plataforma web interactiva que permita la gestión, visualización y análisis de los resultados de las pruebas Saber 11 y Saber Pro, brindando a los usuarios herramientas intuitivas para la consulta de datos, aplicación de filtros analíticos y generación de reportes, a fin de facilitar la interpretación y el aprovechamiento de la información académica.

Objetivos Específicos

1. Desarrollar un sistema de autenticación y control de acceso que gestione el ingreso de usuarios según su rol (administrador, docente o estudiante), garantizando la seguridad y segmentación de las funcionalidades dentro de la plataforma.

2. Diseñar una interfaz de usuario moderna y responsiva mediante el uso de la biblioteca React, que permita una navegación fluida y una experiencia de usuario eficiente en la interacción con los datos académicos.
3. Implementar una base de datos relacional en PostgreSQL que permita el almacenamiento estructurado, seguro y eficiente de los resultados de las pruebas Saber, asegurando la integridad y disponibilidad de la información.
4. Incorporar mecanismos para la carga y validación de datos provenientes de fuentes oficiales, verificando la consistencia y el cumplimiento de los estándares requeridos antes de su almacenamiento.
5. Integrar herramientas de filtrado dinámico y análisis de datos, que posibiliten la exploración de los resultados según criterios como año, institución, programa académico, puntajes, entre otros parámetros relevantes.
6. Desarrollar módulos de generación de reportes gráficos y exportables, que representen la información de forma clara y útil para los distintos usuarios del sistema, facilitando su análisis y aplicación en procesos institucionales.
7. Documentar el diseño y la arquitectura del sistema mediante la elaboración de diagramas UML y de arquitectura, que permitan representar gráficamente los componentes, flujos de información y relaciones funcionales del proyecto.

IV. RESULTADOS

A. Historias de Usuario

Una historia de usuario es una descripción breve sobre una necesidad o funcionalidad desde la perspectiva del usuario final. Se utiliza en metodologías ágiles para definir requisitos y guiar el desarrollo de software de manera iterativa e incremental. Cada historia de usuario incluye un

propósito, criterios de aceptación y una prioridad para facilitar su implementación.

Las historias de usuario presentadas en este informe fueron elaboradas a partir de una previa entrevista y proceso de elicitación con los involucrados en el sistema. Este proceso permitió identificar necesidades clave y definir funcionalidades esenciales para el desarrollo de la plataforma, asegurando que los requisitos reflejen fielmente las expectativas y requerimientos de los usuarios.

Código: HU-01	Usuario: Decano
Título: Creación de Usuarios Manual	
Prioridad: Alta	Tipo: Funcionalidad
<p>Descripción: El sistema debe permitir al decano crear usuarios manualmente. Se desplegará un formulario que debe ser completado con los campos: cédula, el primer nombre, el segundo nombre, el primer apellido, el segundo apellido, email y rol (Decano, director, comité, docente y coordinador saber pro).</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> El decano puede acceder al formulario de creación de usuarios desde su sesión en el sistema. Los campos segundo nombre y segundo apellido son los únicos que no son obligatorios. El sistema valida que no existan usuarios duplicados (por correo o ID). Si hay un duplicado o algún campo vacío, se muestra un mensaje de error. Si todo es válido, el usuario se crea y se muestra un mensaje de confirmación. 	

Tabla 1: historia de usuario Creación de Usuarios Manual.

Código: HU-02	Usuario: Decano
Título: Creación de Usuarios Automática	
Prioridad: Media	Tipo: Funcionalidad
<p>Descripción: El sistema debe permitir al decano la creación automática de usuarios. Esta opción abrirá el explorador de archivos para permitir la carga de un archivo en formato .csv o .xlsx.</p> <p>El archivo tiene que contener estos datos de cada usuario a crear: cédula, el primer nombre, el segundo nombre, el primer apellido, el segundo apellido, email y rol.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> El sistema permite seleccionar un archivo .csv o .xlsx. Todos los campos son obligatorios a excepción de segundo nombre y segundo apellido. El sistema valida que los campos estén completos y no haya usuarios duplicados (por correo o ID). Los usuarios válidos se crean exitosamente. Los usuarios inválidos no se crean y se notifica al decano qué registros fallaron y por qué. 	

Tabla 2: historia de usuario Creación de Usuarios Automática.

Código: HU-03	Usuario: Decano
Título: Listado de Usuarios	
Prioridad: Media	Tipo: Funcionalidad
<p>Descripción: El sistema debe permitir al decano acceder a una vista donde se muestre una tabla con todos los usuarios registrados. Esta lista debe incluir los nombres, apellidos, correo, rol y estado de cada usuario. Además se podrá filtrar por nombre y por rol. Esta funcionalidad facilita el control y monitoreo de los usuarios por parte del decano.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> La tabla permite el desplazamiento si hay muchos usuarios. El sistema actualiza esta lista dinámicamente cuando se registran o modifican usuarios. 	

Tabla 3: historia de usuario Listado de Usuarios.

Código: HU-04	Usuario: Decano
Título: Envío de Credenciales por Correo	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Después de crear un nuevo usuario (de forma manual o automática), el sistema debe generar una contraseña aleatoria y enviársela al correo electrónico registrado del usuario. El mensaje debe incluir su usuario (correo) y la contraseña generada, con una indicación de que debe cambiarla en el primer ingreso al sistema por motivos de seguridad.	
Criterios de aceptación: <ul style="list-style-type: none"> • Una vez creado el usuario, el sistema genera una contraseña aleatoria. • El sistema envía automáticamente un correo al usuario con: su nombre, correo electrónico con el que debe iniciar sesión, la contraseña aleatoria creada, un mensaje indicando que debe de cambiar la contraseña en su primer ingreso. 	

Tabla 4: historia de usuario Envío de Credenciales por Correo.

Código: HU-05	Usuario: Usuarios
Título: Inicio de Sesión	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: Para ingresar al sistema, el usuario debe autenticar su identidad proporcionando su correo electrónico y contraseña. Si alguno de los datos es incorrecto, el acceso será denegado con un mensaje de error. Si las credenciales son válidas, el sistema mostrará el menú principal con las funcionalidades disponibles según el rol asignado al usuario.	
Criterios de aceptación: <ul style="list-style-type: none"> • 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. 	

Tabla 5: historia de usuario Inicio de Sesión.

Código: HU-06	Usuario: Usuarios
Título: Cambio de Contraseña al Primer Inicio de Sesión	
Prioridad: Media	Tipo: Seguridad
Descripción: Cuando un usuario accede por primera vez al sistema, debe cambiar obligatoriamente su contraseña. Al iniciar sesión con la contraseña temporal (enviada por correo), el sistema redirige automáticamente a una pantalla de cambio de contraseña, donde debe ingresar la contraseña actual y definir una nueva.	
Criterios de aceptación: <ul style="list-style-type: none"> • Si es el primer ingreso del usuario, el sistema redirige automáticamente a la pantalla de cambio de contraseña. • En dicha pantalla se solicita: la contraseña actual y la nueva contraseña • El cambio se realiza solo si la contraseña actual es correcta. • La nueva contraseña debe tener un largo mínimo de 6 caracteres • El sistema notifica si el cambio fue exitoso y redirige al menú principal. • Si el cambio no se realiza correctamente, se muestran los errores correspondientes. • El usuario no podrá usar el sistema hasta cambiar la contraseña. 	

Tabla 6: historia de usuario Cambio de Contraseña al Primer Inicio de Sesión.

Código: HU-07	Usuario: Decano, director del programa, coordinador saber pro
Título: Carga de Reportes de Manera Manual	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: El sistema debe permitir a los roles autorizados (decano y director del programa) cargar archivos con resultados Saber Pro en formato .csv o .xlsx. Antes de seleccionar el archivo, el usuario debe ingresar el año y el ciclo académico.	

Criterios de aceptación:	
<ul style="list-style-type: none"> • Solo el decano o el director del programa pueden acceder a esta funcionalidad. • 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. • Si el archivo es válido, los datos se importan y el sistema muestra un mensaje de éxito. • Los datos del reporte cargado se reflejan en el sistema y están disponibles para consulta 	

Tabla 7: historia de usuario Subida de Reportes de Manera Manual.

Código: HU-08	Usuario: Decano, director del programa, coordinador saber pro
Título: Carga de Reportes de Manera Automática	
Prioridad: Media	Tipo: Funcionalidad
Descripción: El sistema debe permitir a los roles autorizados (decano y director del programa) la carga de los resultados de la prueba saber pro de manera automática conectándose a través del ICFES.	
Criterios de aceptación: <ul style="list-style-type: none"> • El sistema ofrece una opción para conectarse al ICFES para obtener los resultados. • Si la carga es exitosa, se muestra un mensaje de confirmación. • Si la carga falla, se informa el motivo del error (autenticación, formato, conexión, etc.). 	

Tabla 8: historia de usuario Subida de Reportes de Manera Automática.

Código: HU-09	Usuario: Usuarios
Título: Visualizar Reportes Subidos	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: El sistema debe permitir a los usuarios ver los resultados de la prueba Saber Pro cargados previamente. Estos se mostrarán en una tabla organizada con columnas. La tabla debe ser clara y estar organizada para facilitar la lectura y el análisis de la información.	
Criterios de aceptación: <ul style="list-style-type: none"> • El sistema muestra los reportes en forma de tabla. • Cada fila de la tabla debe mostrar la información de un estudiante con sus respectivos datos y resultados. • La tabla debe estar ordenada y permitir el desplazamiento vertical si hay muchos registros. • Si no hay datos en la tabla, esta se muestra vacía. • La tabla debe estar actualizada con los últimos reportes cargados, ya sea manual o automáticamente. 	

Tabla 9: historia de usuario Visualizar Reportes Subidos.

Código: HU-10	Usuario: Usuarios
Título: Visualización de Gráfico de Resultados	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: El sistema debe mostrar un gráfico de barras y de líneas que represente visualmente los resultados académicos, específicamente los puntajes promedio por cada módulo evaluado en Saber Pro (como competencias ciudadanas, comunicación escrita, razonamiento cuantitativo, etc.) Este gráfico debe actualizarse automáticamente en función de los datos visibles en la tabla de reportes.	
Criterios de aceptación: <ul style="list-style-type: none"> • El gráfico se genera a partir de los datos actualmente visibles en la tabla de reportes. • Si no hay resultados en la tabla (porque no se han subido reportes o se ha filtrado todo), el gráfico debe mostrarse vacío o con un mensaje indicando que no hay datos para graficar. • Cada barra del gráfico representa el puntaje promedio de un módulo. • El gráfico debe actualizarse automáticamente si se aplican filtros a la tabla (por ejemplo: año, programa, ciclo). 	

Tabla 10: historia de usuario Visualizar Reportes Subidos.

Código: HU-11	Usuario: Usuarios
Título: Filtrar Resultados por Criterios	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: El sistema debe permitir al usuario aplicar filtros sobre la tabla de resultados Saber Pro. Los filtros disponibles son: cédula del estudiante, programa académico y área. También debe existir una opción clara y accesible para eliminar todos los filtros y volver a mostrar la tabla completa.	
Criterios de aceptación: <ul style="list-style-type: none"> El sistema permite aplicar filtros por cédula, programa académico y área. Los filtros se pueden combinar (por ejemplo: cédula + programa). Al aplicar un filtro, la tabla de resultados se actualiza automáticamente. El gráfico de resultados también se actualiza con base en los resultados filtrados. Debe haber un botón para "Eliminar filtros" que restaure la tabla. 	

Tabla 11: historia de usuario Filtrar Resultados por Criterios

Código: HU-12	Usuario: Usuarios
Título: Cambio de Contraseña	
Prioridad: Media	Tipo: Funcionalidad
Descripción: El sistema debe permitir a cualquier usuario cambiar su contraseña de acceso. Para hacerlo, el usuario deberá ingresar su contraseña actual y luego una nueva contraseña.	
Criterios de aceptación: <ul style="list-style-type: none"> El sistema debe validar que la contraseña actual ingresada sea correcta. La nueva contraseña debe tener un mínimo de 6 caracteres para que el cambio sea aceptado. Si los datos ingresados son correctos, el sistema debe actualizar la contraseña y notificar al usuario del éxito del cambio. 	

Tabla 12: historia de usuario Cambio de Contraseña

Código: HU-13	Usuario: Usuarios
Título: Exportar Tabla	
Prioridad: Media	Tipo: Funcionalidad
Descripción: El sistema debe permitir a los usuarios exportar la tabla de resultados actualmente visualizada, ya sea con filtros aplicados o sin ellos. La exportación se debe realizar en formato .xlsx y se debe guardar en el dispositivo del usuario.	
Criterios de aceptación: <ul style="list-style-type: none"> El archivo exportado debe estar en formato .xlsx. Si hay filtros activos, la exportación debe reflejar únicamente los datos filtrados. Si no hay filtros, se exporta toda la tabla completa. Debe haber un botón claramente visible con la opción "Exportar" o similar, ubicado cerca de la tabla. El archivo debe descargarse correctamente en el dispositivo del usuario. 	

Tabla 13: historia de usuario Exportar Tabla

Código: HU-14	Usuario: Decano, director de programa, comité del programa
Título: Crear acción de mejora	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: El sistema debe permitir a los usuarios con rol de decano, miembro del comité o director de programa crear una acción de mejora a un reporte en específico. La acción debe incluir al menos una descripción. No se pueden asociar varias acciones a un mismo reporte desde esta opción.	
Criterios de aceptación: <ul style="list-style-type: none"> El sistema permite a usuarios con rol de decano, miembro del comité o director de programa acceder a la opción de crear acciones de mejora. El formulario de acción de mejora debe solicitar 	

<p>una descripción.</p> <ul style="list-style-type: none"> La acción de mejora solo puede ser asociada a muchos reportes. Si no se completan todos los campos obligatorios, el sistema mostrará un mensaje de error y no permitirá guardar la acción. Al guardar correctamente, la acción de mejora queda visible en el reporte correspondiente.

Tabla 14: historia de usuario Crear Acción de Mejora

<p>desactivar usuarios que ya han sido registrados, para controlar el acceso al sistema según sea necesario.</p>
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> El decano puede cambiar el estado de un usuario con una acción simple (botón o interruptor). Un usuario desactivado no puede iniciar sesión. No se elimina al usuario, solo se inactiva.

Código: HU-15	Usuario: Decano, director de programa, miembro del comité
Título: Análisis de IA para acción de mejora	
Prioridad: Alta	Tipo: Funcionalidad
<p>Descripción: El sistema debe permitir a los usuarios con rol de decano, miembro del comité o director de programa crear una acción de mejora a un reporte en específico. La acción debe incluir al menos una descripción. No se pueden asociar varias acciones a un mismo reporte desde esta opción.</p>	
<p>Criterios de aceptación:</p> <ul style="list-style-type: none"> El sistema debe enviar la acción de mejora creada al motor de análisis de Google IA Studio. El resultado del análisis debe mostrarse visualmente junto con la acción de mejora evaluada. Solo usuarios autorizados (decano, miembro del comité o director de programa) pueden ver el análisis. 	

Tabla 15: historia de usuario Análisis de IA para acción de mejora

Tabla 16: historia de usuario Activación o Desactivación de Usuarios.

B. Requerimientos Funcionales

Los requerimientos funcionales describen las acciones que el sistema debe realizar para cumplir con las necesidades del usuario. A continuación, se detallan en tablas individuales:

Identificación del Requerimiento	RF01
Nombre del requerimiento	Creación manual de usuarios
Descripción del Requerimiento	El sistema le debe permitir al decano la creación de usuarios a través de un formulario.
Requerimiento no Funcional	RNF05
Prioridad del Requerimiento: Alta	

Tabla 17: requerimiento funcional para registro manual de usuario

Código: HU-16	Usuario: Decano
Título: Activación o Desactivación de Usuarios	
Prioridad: Alta	Tipo: Funcionalidad
Descripción: El decano tendrá el poder activar o	

Identificación del Requerimiento	RF02
Nombre del requerimiento	Creación automática de usuarios
Descripción del Requerimiento	El sistema le debe permitir al decano la creación de usuarios de manera automática al cargar un archivo con los datos de los usuarios a crear.
Requerimiento no Funcional	RNF04, RNF05
Prioridad del Requerimiento: Media	

Tabla 18: requerimiento funcional para registro automático de usuarios

Identificación del Requerimiento	RF03
Nombre del requerimiento	Envío automático de credenciales por correo
Descripción del Requerimiento	Tras la creación de un usuario el sistema mandará un correo al usuario registrado con su clave de acceso.
Requerimiento no Funcional	RNF01
Prioridad del Requerimiento: Alta	

Tabla 19: requerimiento funcional para el envío de credenciales por correo

Identificación del Requerimiento	RF04
Nombre del requerimiento	Inicio de sesión
Descripción del Requerimiento	El sistema debe permitir al usuario autenticarse mediante un formulario de correo y contraseña.
Requerimiento no Funcional	RNF02, RNF03
Prioridad del Requerimiento: Alta	

Tabla 20: requerimiento funcional para el inicio de sesión

Identificación del Requerimiento	RF05
Nombre del requerimiento	Cambio de contraseña
Descripción del Requerimiento	El sistema debe permitir al usuario el cambio de su contraseña
Requerimiento no Funcional	RNF01, RNF02
Prioridad del Requerimiento: Alta	

Tabla 21: requerimiento funcional para el cambio de contraseña

Identificación del Requerimiento	RF06
Nombre del requerimiento	Listar Usuarios
Descripción del Requerimiento	El sistema debe mostrar al decano una tabla con todos los usuarios registrados junto con sus datos
Requerimiento no Funcional	RNF06
Prioridad del Requerimiento: Media	

Tabla 22: requerimiento funcional para listado de usuarios

Identificación del Requerimiento	RF07
Nombre del requerimiento	Carga manual de reportes
Descripción del Requerimiento	El decano, director de programa y coordinador saber pro, tienen la capacidad de cargar reportes de manera manual.
Requerimiento no Funcional	RNF04
Prioridad del Requerimiento: Alta	

Tabla 23: requerimiento funcional para el carga manual de reportes

Identificación del Requerimiento	RF08
Nombre del requerimiento	Carga automática de reportes
Descripción del Requerimiento	El decano, director de programa y coordinador saber pro, serán capaces de cargar usuarios de un archivo para su creación automática
Requerimiento no Funcional	RNF04, RNF06
Prioridad del Requerimiento: Media	

Tabla 24: requerimiento funcional para el carga automática de reportes

Identificación del Requerimiento	RF09
Nombre del requerimiento	Visualizar tabla de resultados
Descripción del Requerimiento	Los usuarios podrán ver en una tabla los resultados de la prueba saber pro subidos hasta el momento.
Requerimiento no Funcional	RNF06, RNF07
Prioridad del Requerimiento: Alta	

Tabla 25: requerimiento funcional para la visualización de resultados

Identificación del Requerimiento	RF10
Nombre del requerimiento	Visualizar gráfico de resultados
Descripción del Requerimiento	Los usuarios podrán ver la representación de las tablas en forma de gráfico.
Requerimiento no Funcional	RNF06
Prioridad del Requerimiento: Alta	

Tabla 26: requerimiento funcional para la visualización de gráfico de resultados

Identificación del Requerimiento	RF11
Nombre del requerimiento	Filtrado de resultados
Descripción del Requerimiento	Los usuarios podrán aplicar filtros a la tabla de resultados, estos filtros también modificarán el gráfico.
Requerimiento no Funcional	RNF06
Prioridad del Requerimiento: Alta	

Tabla 27: requerimiento funcional para el filtrado de resultados

Identificación del Requerimiento	RF12
Nombre del requerimiento	Exportar tabla de resultados
Descripción del Requerimiento	Los usuarios podrán exportar la tabla de resultados visualizada (con o sin filtros), esto en archivo .xlsx, descargándolo en el dispositivo.
Requerimiento no Funcional	RNF04, RNF07
Prioridad del Requerimiento: Media	

Tabla 28: requerimiento funcional para la exportación de tablas de resultados

Identificación del Requerimiento	RF13
Nombre del requerimiento	Gestión de acciones de mejora
Descripción del Requerimiento	El sistema debe permitir a los usuarios autorizados crear una acción de mejora asociada a un único reporte de resultados Saber Pro.
Requerimiento no Funcional	RNF03
Prioridad del Requerimiento: Alta	

Tabla 29: requerimiento funcional para la gestión de acciones de mejora

Identificación del Requerimiento	RF14
Nombre del requerimiento	Análisis de acción de mejora mediante IA
Descripción del Requerimiento	El sistema debe permitir que los usuarios con acceso, soliciten un análisis de efectividad sobre una acción de mejora registrada, utilizando un modelo de inteligencia artificial. El sistema mostrará al usuario una evaluación sobre si dicha acción fue efectiva o no, junto con una breve justificación.
Requerimiento no Funcional	RNF08
Prioridad del Requerimiento: Media	

Tabla 30: requerimiento funcional para el análisis de IA

C. Requerimientos no Funcionales

Los requerimientos no funcionales definen las características del sistema en cuanto a seguridad, rendimiento, usabilidad, disponibilidad y compatibilidad.

Identificación del Requerimiento:	RNF01
Nombre del Requerimiento:	Seguridad de almacenamiento
Descripción del Requerimiento:	La información sensible (contraseñas, datos personales) debe almacenarse cifrada.
Prioridad del Requerimiento: Alta	

Tabla 31: requerimiento no funcional para la seguridad de almacenamiento

Identificación del Requerimiento:	RNF02
Nombre del Requerimiento:	Autenticación segura
Descripción del Requerimiento:	El sistema debe validar credenciales de forma segura (hashing de contraseñas) y rechazar accesos no válidos.
Prioridad del Requerimiento: Alta	

Tabla 32: requerimiento no funcional para la autenticación segura

Identificación del Requerimiento:	RNF03
Nombre del Requerimiento:	Control de acceso por roles
Descripción del Requerimiento:	El sistema debe dar acceso a funcionalidades según el rol del usuario.
Prioridad del Requerimiento: Alta	

Tabla 33: requerimiento no funcional para el control de acceso por roles

Identificación del	RNF04
--------------------	-------

Requerimiento:	
Nombre del Requerimiento:	Compatibilidad de formatos de archivo
Descripción del Requerimiento:	El sistema solo aceptará archivos .csv y .xlsx para la carga de datos.
Prioridad del Requerimiento: Alta	

Tabla 34: requerimiento no funcional para la compatibilidad de formato de archivos

Identificación del Requerimiento:	RNF05
Nombre del Requerimiento:	Validación de unicidad de usuarios
Descripción del Requerimiento:	El sistema debe impedir la creación de usuarios con cédula duplicada.
Prioridad del Requerimiento: Alta	

Tabla 35: requerimiento no funcional para la unicidad de usuario

Identificación del Requerimiento:	RNF06
Nombre del Requerimiento:	Rendimiento en carga y visualización
Descripción del Requerimiento:	La carga de reportes (tabla y gráfica) y las operaciones de filtrado no deben demorar más de 3 segundos.
Prioridad del Requerimiento: Media	

Tabla 36: requerimiento no funcional para el rendimiento de la carga

Identificación del	RNF07
--------------------	-------

Requerimiento:	
Nombre del Requerimiento:	Usabilidad y accesibilidad
Descripción del Requerimiento:	La interfaz debe ser intuitiva, responsive y permitir desplazamientos fluidos en tablas con muchos registros.
Prioridad del Requerimiento: Media	

Tabla 37: requerimiento no funcional para la usabilidad y accesibilidad

Identificación del Requerimiento:	RNF08
Nombre del Requerimiento:	Integración con servicios externos de IA
Descripción del Requerimiento:	El sistema debe estar integrado de forma segura y eficiente con un servicio externo de inteligencia artificial, garantizando tiempos de respuesta adecuados, confidencialidad de los datos transmitidos y disponibilidad mínima del 99% del servicio.
Prioridad del Requerimiento: Media	

Tabla 33: requerimiento no funcional para el control de acceso por roles

V. CONCLUSIÓN

El desarrollo de la plataforma web para la gestión y visualización de los resultados de las pruebas Saber 11 y Saber Pro representa un aporte significativo al fortalecimiento de los procesos académicos y administrativos en las instituciones educativas. A través de la implementación de tecnologías modernas como Spring Boot para el frontend y PostgreSQL para la gestión de datos, fue posible construir una herramienta robusta, interactiva y accesible, que permite a los usuarios consultar, analizar y comprender la información académica de manera eficiente.

La integración de funcionalidades clave como el inicio de sesión con control de acceso, la carga y validación de datos, el uso de filtros dinámicos, la generación de gráficos interactivos y la exportación de reportes, demuestra una solución integral que

responde a las necesidades reales del entorno educativo. Asimismo, la documentación mediante diagramas de arquitectura y modelos UML permitió estructurar el sistema de forma coherente, facilitando su comprensión, mantenimiento y escalabilidad.

En definitiva, este proyecto no solo automatiza y optimiza el manejo de los resultados de las pruebas Saber, sino que también promueve el uso de herramientas tecnológicas en la toma de decisiones informadas, apoyando el mejoramiento continuo de la calidad educativa.

Referencias

- Instituto Colombiano para la Evaluación de la Educación – ICFES. (s.f.). *Pruebas Saber 11 y Saber Pro*. Recuperado de: <https://www.icfes.gov.co>
- PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*. Recuperado de: <https://www.postgresql.org/docs/>
- Object Management Group (OMG). (2017). *Unified Modeling Language (UML) Specification*. Recuperado de: <https://www.omg.org/spec/UML/>
- Microsoft. (2023). *Introduction to CQRS*. Recuperado de: <https://learn.microsoft.com/en-us/azure/architecture/patterns/cqrs>