

Especificación de Requisitos del Software (ERS)

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

Resumen

El presente documento detalla la Especificación de Requisitos del Software (ERS) para el desarrollo de una plataforma web destinada a la gestión, consulta y análisis de los resultados de las pruebas Saber 11 y Saber Pro, aplicadas por el ICFES en Colombia. El sistema está diseñado para ser utilizado por distintos tipos de usuarios —administradores, docentes y estudiantes—, cada uno con funcionalidades específicas, como la carga de archivos, filtrado de resultados, generación de reportes y visualización de gráficos interactivos.

El backend de la aplicación ha sido implementado utilizando el framework Spring Boot, mientras que el almacenamiento de datos se gestiona mediante el sistema de bases de datos relacional PostgreSQL. La comunicación entre cliente y servidor se realiza a través de servicios REST, bajo el protocolo HTTPS y empleando autenticación basada en tokens (JWT), garantizando así la seguridad y el control de acceso.

El documento incluye tanto los requisitos funcionales como no funcionales, las interfaces externas, las restricciones técnicas, y los atributos de calidad que debe cumplir el sistema. Se describen las funciones principales, los casos de uso previstos, las características de los usuarios, así como los aspectos de rendimiento, seguridad y mantenibilidad. Además, se identifican las dependencias tecnológicas y se establecen lineamientos para futuras mejoras.

En resumen, este documento proporciona una base sólida y estructurada para el desarrollo, validación y mantenimiento del sistema, asegurando que cumpla con las necesidades de los usuarios y los estándares técnicos esperados.

Contenido

1. Introducción.....	3
1.1 Propósito del documento	3
1.2 Alcance del sistema	3
1.3 Definiciones, acrónimos y abreviaturas.....	4
1.4 Referencias	4
2. Descripción General	5
2.1 Perspectiva del producto.....	5
2.2 Funciones del producto.....	5
2.3 Característica del usuario.....	6
2.4 Restricciones.....	6
2.5 Suposiciones y dependencias	7
3. Requisitos Específicos.....	7
3.1 Requisitos funcionales.....	7
3.2 Requisitos no funcionales.....	9
3.3 Interfaces externas	9
4. Interfaces Externas	10
4.1 Interfaz de Usuario	10
4.2 Interfaz del Sistema	10
4.3 Interfaz de Base de Datos	10
4.4 Interfaz de Comunicación.....	11

1. Introducción

1.1 Propósito del documento

El propósito de este documento es presentar de forma estructurada, clara y completa los requisitos del software desarrollado para la plataforma de visualización y análisis de los resultados de las pruebas Saber Pro, realizadas por el ICFES en Colombia. Esta especificación tiene como objetivo principal establecer un marco de referencia común para todos los involucrados en el ciclo de vida del proyecto, incluyendo desarrolladores, evaluadores, administradores del sistema, usuarios finales, y cualquier parte interesada en el diseño, implementación, prueba, despliegue y mantenimiento del sistema.

El documento incluye tanto los **requisitos funcionales** (que definen el comportamiento del sistema ante entradas específicas o eventos) como los **requisitos no funcionales** (que imponen restricciones sobre aspectos como el rendimiento, la seguridad, la escalabilidad y la mantenibilidad). Además, se presentan las interfaces externas, restricciones, características de los usuarios, y otras consideraciones relevantes.

Este documento también busca servir como guía para futuras actualizaciones del sistema, facilitando su comprensión técnica y evolución mediante una documentación estandarizada.

1.2 Alcance del sistema

El sistema desarrollado consiste en una aplicación orientada a la Universidad de los Llanos, docentes y directivos, que permite la **consulta estructurada, filtrada y visual** de los resultados obtenidos en las pruebas Saber Pro.

La plataforma ofrece funcionalidades clave como:

- Carga de archivos CSV que contienen los resultados oficiales suministrados por el ICFES.
- Carga de reportes a través de la conexión con Datos Libres del gobierno.
- Segmentación y filtrado por cedula, programa académico y área
- Visualización de los datos mediante gráficos y tablas organizadas.
- Exportación de los resultados procesados en formatos útiles para su análisis offline.
- Acceso controlado mediante autenticación según el tipo de usuario (decano, director de programa, comité de programa, coordinador saber pro y profesor).
- Creación y gestión de usuarios para mayor confidencialidad en los datos.
- Creación de acciones de mejora con el fin de mejorar el rendimiento de los estudiantes en las pruebas.

El sistema está diseñado con una arquitectura modular. El **backend está implementado con Spring Boot**, lo que garantiza una base robusta en Java para la creación de servicios RESTful, mientras que la gestión y persistencia de datos se realiza utilizando **PostgreSQL**, una base de datos relacional potente y ampliamente adoptada.

El sistema se accede a través de una aplicación de escritorio, diseñada para ofrecer una experiencia intuitiva y eficiente para sus distintos perfiles de usuario.

1.3 Definiciones, acrónimos y abreviaturas

A continuación, se presentan los términos y acrónimos empleados frecuentemente en este documento:

- **ERS:** Especificación de Requisitos del Software. Documento técnico que define qué debe hacer el sistema y cómo debe comportarse.
- **Saber 11 / Saber Pro:** Pruebas estandarizadas nacionales aplicadas por el ICFES para evaluar la calidad de la educación media y superior en Colombia.
- **ICFES:** Instituto Colombiano para la Evaluación de la Educación.
- **Spring Boot:** Framework para el desarrollo de aplicaciones Java orientadas a microservicios, APIs REST y aplicaciones web.
- **PostgreSQL:** Sistema de gestión de bases de datos relacional, robusto y de código abierto.
- **REST (Representational State Transfer):** Estilo arquitectónico usado para la creación de servicios web basados en recursos accesibles mediante métodos HTTP.
- **JWT (JSON Web Token):** Estándar para transmitir información segura entre partes como un objeto JSON firmado.
- **Usuario:** Persona que interactúa con el sistema. Se clasifica en:
 - **Decano**
 - **Coordinador saber pro**
 - **Director de programa**
 - **Comité de programa**
 - **Profesor**

1.4 Referencias

- Instituto Colombiano para la Evaluación de la Educación (ICFES). Sitio oficial: <https://www.icfes.gov.co>
- Spring Boot (2024). Proyecto oficial: <https://spring.io/projects/spring-boot>

- PostgreSQL. Documentación oficial: <https://www.postgresql.org/docs/>
- IEEE Std 830-1998. *Recommended Practice for Software Requirements Specifications*. Institute of Electrical and Electronics Engineers.

2. Descripción General

2.1 Perspectiva del producto

La plataforma se concibe como una **aplicación modular**, diseñada para ofrecer acceso estructurado a los datos de las pruebas Saber Pro mediante una interfaz clara y segmentada por tipo de usuario. El sistema se basa en una arquitectura cliente-servidor, donde:

- El **backend**, desarrollado con **Spring Boot**, se encarga de exponer servicios web a través de una API REST, realizar la lógica de negocio, validar y procesar los datos, y gestionar el acceso a la base de datos.
- La **base de datos PostgreSQL** actúa como repositorio centralizado de toda la información cargada, incluyendo resultados de pruebas, información de usuarios y registros de actividad.
- El **frontend**, aunque no se detalla en este documento, se conecta a través de llamadas HTTP a los endpoints REST del backend para representar los datos de manera visual y responsiva.

El producto no es una solución genérica, sino que está específicamente adaptado a los formatos de datos proporcionados por el ICFES y a las necesidades de análisis de usuarios educativos.

2.2 Funciones del producto

El sistema implementa las siguientes funcionalidades principales:

- **Autenticación y control de acceso:** El sistema requiere que cada usuario inicie sesión, y según su tipo de cuenta, se le habilitan diferentes funcionalidades.
- **Carga de datos:** Los roles permitidos pueden cargar archivos CSV generados por el ICFES que contienen los resultados detallados de las pruebas Saber Pro. El sistema realiza una validación automática de estos archivos antes de almacenarlos.
- **Validación de datos:** Se validan tanto la estructura del archivo como la coherencia de los datos.
- **Filtrado dinámico:** Los usuarios pueden aplicar filtros personalizados sobre los resultados cargados, permitiendo consultar por cédula del estudiante, programa académico y área.
- **Visualización interactiva:** Los resultados pueden visualizarse en formato tabular o mediante gráficos, facilitando el análisis comparativo y el entendimiento de las tendencias.

- **Exportación de reportes:** Los usuarios tienen la opción de generar reportes con los resultados filtrados, los cuales pueden ser descargados para su análisis posterior.

Estas funciones están disponibles a través de una interfaz accesible, y todas las operaciones críticas están respaldadas por servicios REST que aseguran una separación adecuada entre presentación y lógica de negocio.

2.3 Característica del usuario

El sistema contempla tres perfiles de usuario, cada uno con accesos y capacidades diferenciadas:

- **Decano:** Super usuario, tiene acceso a todas las funcionalidades del sistema, estas son: Gestión de usuarios, carga de reportes, visualización de resultados, creación de acciones de mejora.
- **Coordinador saber pro:** Usuario con capacidad para la carga de reportes, y visualización de resultados.
- **Director de programa:** Usuario con capacidad para la carga de reportes, creación de acciones de mejora y visualización de reportes.
- **Comité de programa:** Usuario con capacidad para la creación de acciones de mejora y visualización de resultados.
- **Profesor:** Usuario que solo puede visualizar reportes.

Cada tipo de usuario accede a un subconjunto de funcionalidades específicas, lo cual está regulado mediante un sistema de autenticación y control de permisos implementado en el backend.

2.4 Restricciones

El desarrollo y funcionamiento del sistema están sujetos a las siguientes restricciones:

- El **backend debe estar construido utilizando Spring Boot**, manteniendo un diseño modular basado en servicios REST.
- Toda la **persistencia de datos debe realizarse en PostgreSQL**, utilizando estructuras relacionales y claves foráneas para asegurar la integridad.
- El sistema debe estar **protegido mediante autenticación**, y cada operación sensible debe ser accesible solo con los permisos adecuados.
- Los archivos cargados deben tener el **formato CSV con estructura definida** según lo proporcionado por el ICFES.

2.5 Suposiciones y dependencias

Durante el desarrollo del sistema, se consideraron las siguientes suposiciones y dependencias como válidas:

- Los archivos de resultados provienen directamente del ICFES y cuentan con **una estructura estable y verificable**.
- El sistema se desplegará en un servidor que soporte entornos Java, tenga acceso a un motor de base de datos PostgreSQL y permita conexiones HTTPS.
- Se asume que los usuarios tienen conocimientos básicos de navegación web y están familiarizados con los resultados de las pruebas Saber.

3. Requisitos Específicos

En esta sección se describen en detalle los requisitos funcionales y no funcionales que definen el comportamiento esperado del sistema. También se presentan las interfaces externas, restricciones de diseño, atributos del sistema y otras especificaciones técnicas relevantes.

3.1 Requisitos funcionales

A continuación, se detallan los comportamientos esperados del sistema bajo distintas condiciones. Cada requisito está numerado y redactado de manera que permita su verificación durante las etapas de prueba.

- **RF01 – Creación manual de usuarios:**
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).
- **RF02 – Creación de Usuarios Automática:**
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. 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.
- **RF03 – Envío de Credenciales por Correo:**
Después de crear un nuevo usuario, 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.
- **RF04 – Inicio de sesión:**
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.

- **RF05 – Cambio de contraseña:**

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.

- **RF06 – Listar Usuarios:**

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.

- **RF07 – Carga manual de reportes:**

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.

- **RF08 – Carga automática de reportes:**

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.

- **RF09 – Visualizar tabla de resultados:**

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.

- **RF10 – Visualizar gráfico de resultados:**

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.

- **RF11 – Filtrado de resultados:**

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.

- **RF12 – Exportar tabla de resultados:**

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.

3.2 Requisitos no funcionales

Estos requisitos definen propiedades de calidad del sistema que no se relacionan con funciones específicas, pero son esenciales para su desempeño, seguridad y mantenibilidad.

- **RNF01 – Seguridad de almacenamiento:**
La información sensible (contraseñas, datos personales) debe almacenarse cifrada.
- **RNF02 – Autenticación segura:**
El sistema debe validar credenciales de forma segura (hashing de contraseñas) y rechazar accesos no válidos.
- **RNF03 – Control de acceso por roles:**
El sistema debe dar acceso a funcionalidades según el rol del usuario.
- **RNF04 – Compatibilidad de formatos de archivo:**
El sistema solo aceptará archivos .csv y .xlsx para la carga de datos.
- **RNF05 – Validación de unicidad de usuarios:**
El sistema debe impedir la creación de usuarios con cédula duplicada.
- **RNF06 – Rendimiento en carga y visualización:**
La carga de reportes (tabla y gráfica) y las operaciones de filtrado no deben demorar más de 3 segundos.

3.3 Interfaces externas

Esta sección describe cómo interactúa el sistema con otros componentes o sistemas externos.

- **Interfaz de usuario:**
Se espera que la interfaz cliente consuma los servicios REST expuestos por el backend. Esta interfaz debe ser capaz de realizar autenticación, enviar filtros, recibir resultados en JSON, y representar los datos adecuadamente.
- **Interfaz del sistema (backend):**
El backend está construido con Spring Boot, estructura sus funcionalidades en controladores, servicios y repositorios. Expone una API REST accesible mediante peticiones HTTP.
- **Interfaz de base de datos:**
El sistema utiliza PostgreSQL para la gestión de la información. La conexión a la base de datos se realiza a través de un pool de conexiones administrado por Spring Data JPA.
- **Interfaz de comunicación:**
Todo el intercambio de información entre cliente y servidor se realiza a través del protocolo HTTPS. Se emplea autenticación por tokens para proteger cada petición (autenticación basada en JWT).

4. Interfaces Externas

4.1 Interfaz de Usuario

La interfaz de usuario es una aplicación de escritorio que se conecta al backend a través de HTTP. Permite:

- Iniciar sesión según el rol.
- Cargar archivos CSV.
- Aplicar filtros de búsqueda.
- Visualizar resultados en tabla y gráficos.
- Exportar reportes en Excel.
- Crear usuarios.
- Crear acciones de mejora.

4.2 Interfaz del Sistema

El backend está desarrollado en Spring Boot y expone servicios REST. Las operaciones están organizadas por recursos (usuarios, resultados, reportes). La documentación de la API puede seguir el estándar OpenAPI (Swagger).

4.3 Interfaz de Base de Datos

Se utiliza PostgreSQL como sistema de gestión. La conexión se realiza mediante JPA (Java Persistence API). Las tablas incluyen:

- Users
- Personas
- Roles
- Cambio_contraseña
- Programas académicos
- Catalogo_modulos
- Acciones_mejoras
- Evaluados
- Ciudades

- Grupos_referencias
- Tipos_documentos
- Ciudades
- Tipos_evaluados
- Resultados_modulos
- Resultados_globales
- Registros_evaluaciones

4.4 Interfaz de Comunicación

Todo el intercambio entre cliente y servidor debe realizarse a través del protocolo HTTPS. Las solicitudes están protegidas mediante autenticación JWT (JSON Web Tokens).