

Sistema de pruebas en línea

Presentado por:

Pablo Andrés Nino Barreto- pninob@unal.edu.co

Jeisson Duván Bareño Ruiz- jbareno@unal.edu.co

Víctor Camilo Cañón Castellanos - vcanonc@unal.edu.co

Jacobo Álzate Corredor- jalzateco@unal.edu.co

Profesor:

Oscar Eduardo Alvarez Rodriguez- oalvarezr@unal.edu.co



Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas e Industrial

2025

Índice

Proceso de consenso sobre el tema del proyecto.....	4
Alcance del proyecto.....	6
Funcionalidades principales del sistema.....	7
<i>Características principales para el administrador.....</i>	7
<i>Características principales para el docente.....</i>	7
<i>Características principales para el estudiante.....</i>	8
Requerimientos funcionales.....	9
Requerimientos no funcionales.....	11
Análisis de requerimientos.....	12
<i>Priorización.....</i>	12
<i>Estimación de esfuerzo.....</i>	13
Análisis de gestión del desarrollo.....	16
<i>Tiempo.....</i>	16
<i>Costos.....</i>	16
<i>Producto mínimo viable (MVP).....</i>	17
Referencias.....	19

Sistema de pruebas en línea

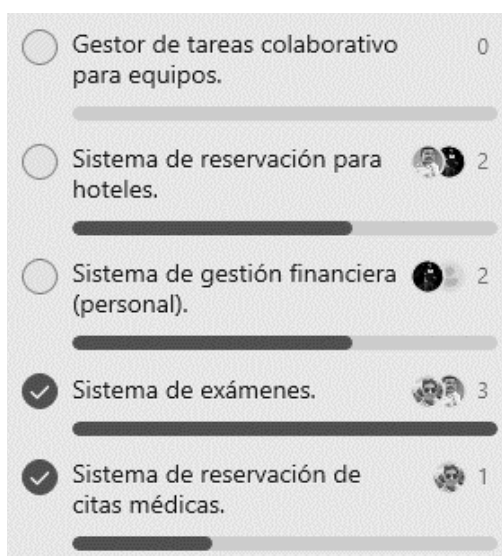
Este documento tiene como objetivo la recopilación, análisis y especificación de los requerimientos de un sistema de exámenes para un colegio como proyecto de la materia de Ingeniería de Software I.

Proceso de consenso sobre el tema del proyecto

Se realizó una reunión donde se obtuvieron las siguientes opciones a través de una lluvia de ideas previa por cada integrante del equipo:

1. Gestor de tareas colaborativo para equipos de trabajo.
2. Sistema de reservas para un hotel.
3. Sistema de gestión de las finanzas personales.
4. Sistema de aplicación de exámenes.
5. Reservaciones de citas médicas para una clínica.
6. Sistema de gestión bibliotecaria.
7. Sistema de intercambio de libros.

Se discutieron cada una de las opciones descartando algunas y finalmente aplicamos una encuesta grupal, en donde cada integrante deberá elegir dos ideas que preferiría desarrollar:



Por lo tanto, se eligió la idea “Sistema de aplicación de exámenes” para la formulación y ejecución del proyecto descrito en este documento.

Contexto

No existen sistemas enfocados únicamente en la aplicación de pruebas en instituciones educativas que hagan el proceso de recopilación centralizada de datos para realizar un análisis de los resultados académicos más óptimo para el sistema de educación colombiano, especialmente en el sector público, es decir, colegios distritales. Por lo que buscamos formular un sistema de evaluación básico que se ajuste a las particularidades de la organización y niveles de la educación básica secundaria, y educación media que permita optimizar la creación, aplicación y calificación de exámenes en dichos entornos escolares, con el fin de reducir los procesos manuales y errores humanos que conlleva la utilización de métodos basados en lápiz y papel, además de proveer herramientas para el análisis del rendimiento dada la necesidad de mecanismos eficientes para evaluar y garantizar la calidad educativa priorizando la reducción de desigualdades y la mejora de procesos educativos mediante herramientas innovadoras, objetivo fijado en el Plan Estratégico Institucional (PEI) del Ministerio de Educación Nacional y el Decreto 1075 de 2015.

Por ello el sistema contemplará la creación de preguntas y exámenes por parte de los profesores, la aplicación de estos a grupos específicos de estudiantes, calificación automática, generación de reportes, y acceso de estudiantes a los resultados de sus evaluaciones. Esto permitirá reducir el tiempo invertido en la preparación y calificación, garantizar la transparencia y precisión en las evaluaciones, y centralizar la gestión de exámenes en una plataforma única.

Alcance del proyecto

Desarrollar e implementar un sistema integral de gestión de exámenes en línea que permita a los profesores crear, administrar y calificar exámenes, así como a los estudiantes presentarlos y consultar sus resultados mejorando la eficiencia en los procesos de evaluación y permitiendo un análisis detallado del rendimiento académico en colegios.

Funcionalidades principales del sistema

A continuación, se enlistan las características y funcionalidades proyectadas para el software, segmentadas por los roles que tendrá el sistema:

Características principales para el administrador

- 1. Gestión de grupos:** El administrador podrá crear, editar, y eliminar grupos de estudiantes.
- 2. Gestión de usuarios:** El sistema permitirá crear, consultar y eliminar datos de los usuarios, asignarles un rol (Administrador, Docente, Estudiante) para los cuales podrá realizar las siguientes acciones:
 - a.** Docentes: Asignarlos a materias, grados y grupos.
 - b.** Estudiantes: Solo se podrá relacionar con un grado y grupo.
- 3. Supervisión de evaluaciones:** El administrador solo podrá consultar las evaluaciones cargadas en el sistema.
- 4. Generación de credenciales de usuarios:** El administrador podrá generar y eliminar credenciales para los usuarios registrados.
- 5. Gestión del banco de preguntas:** El administrador podrá consultar, modificar y eliminar preguntas del banco de preguntas.
- 6. Exportar calificaciones:** El sistema permitirá exportar los resultados de exámenes por grupos, materias o estudiantes en formato excel.
- 7. Generación de reportes:** El sistema podrá generar reportes del rendimiento académico por grupo, materia y grado.

Características principales para el docente

- 1. Gestión de usuario:** El docente podrá gestionar la información relacionada a su cuenta.
- 2. Gestión de evaluaciones:** El docente diseñará exámenes cargando preguntas en distintos formatos, configurando plazos y temporizadores de tiempo para la aplicación de los mismos, número de intentos permitidos y criterios de calificación; y los

asignará a los grupos a su cargo. También podrá ver todas las evaluaciones aplicadas del grupo y materia que tiene asignados.

3. **Gestión de banco de preguntas:** El sistema permitirá al docente agregar y eliminar las preguntas cargadas de sus exámenes al banco de preguntas de su materia y el grado al cual pertenece el examen.
4. **Reactivación de evaluaciones:** Se permitirá reactivar los plazos de evaluaciones aunque ya hayan sido aplicadas, para grupos o individuos.
5. **Exportar calificaciones:** El sistema permitirá exportar los resultados de exámenes por grupos, materias o estudiantes en formato excel.
6. **Generación de reportes:** El sistema solo le permitirá al docente generar reportes del rendimiento académico por grupo y estudiante.

Características principales para el estudiante

1. **Realización de evaluaciones:** El estudiante podrá ingresar y realizar las evaluaciones que tiene asignadas dado el grupo y grado al que pertenece.
2. **Visualización de resultados:** El sistema permitirá al estudiante consultar el historial de evaluaciones aplicadas y sus respectivas calificaciones.
3. **Seguimiento de progreso:** El estudiante podrá visualizar estadísticas sobre su rendimiento académico a lo largo del tiempo.
4. **Notificaciones:** Envío de alertas sobre próximas evaluaciones, reactivaciones de plazos, o disponibilidad de resultados.

Requerimientos funcionales

RF-01. Los grupos de estudiantes podrán gestionarse mediante la creación, edición y eliminación de los mismos.

RF-02. Los usuarios podrán ser gestionados a través de la creación, consulta y eliminación de sus datos, además de la asignación de roles como Administrador, Docente o Estudiante.

RF-03. Las materias, grados y grupos podrán asignarse directamente a los docentes registrados.

RF-04. Los estudiantes se relacionarán con un único grado y grupo al momento de su registro.

RF-05. Las evaluaciones podrán ser creadas, modificadas y eliminadas antes de su aplicación, incluyendo la configuración de preguntas en distintos formatos, plazos y temporizadores, número de intentos permitidos y criterios de calificación.

RF-06. Será posible asignar evaluaciones a grupos específicos de acuerdo con el grado y la materia correspondiente.

RF-07. Los administradores tendrán acceso solo para supervisar y consultar las evaluaciones creadas y aplicadas, mientras que los docentes podrán gestionarlas según sus asignaciones.

RF-08. El banco de preguntas incluirá funciones para agregar preguntas desde las evaluaciones creadas, organizarlas por materia y grado, y permitir su modificación o eliminación.

RF-09. Los plazos de evaluación podrán reactivarse para estudiantes o grupos específicos, incluso después de haber sido aplicadas.

RF-10. Los usuarios podrán autenticarse utilizando credenciales generadas previamente por el administrador, y los docentes tendrán la posibilidad de gestionar la información de su cuenta, como actualizar datos personales o cambiar contraseñas.

RF-11. Las calificaciones de exámenes podrán exportarse en formato Excel, permitiendo la selección de datos por grupo, materia o estudiante.

RF-12. Los reportes de rendimiento académico incluirán estadísticas básicas, organizadas por grupo, materia, grado o estudiante.

RF-13. Los estudiantes podrán acceder y realizar las evaluaciones asignadas a su grupo y grado correspondiente.

RF-14. Los resultados de las evaluaciones aplicadas estarán disponibles para los estudiantes, incluyendo un historial completo con las calificaciones obtenidas.

RF-15. Los estudiantes recibirán notificaciones automáticas con recordatorios sobre próximas evaluaciones, mostrando fechas y horarios límite para su realización, tanto en la interfaz del sistema como mediante WhatsApp.

RF-16. Cada estudiante tendrá acceso a un módulo de seguimiento donde podrá visualizar un resumen de su desempeño académico.

RF-17. El sistema permitirá a los docentes agregar preguntas con contenido multimedia.

Requerimientos no funcionales

RNF-01. Diseño responsive: El sistema tendrá soporte para pantallas de móviles y de escritorio.

RNF-02. Interfaz simple: La interfaz debe ser simple, con un menú claro y opciones fácilmente accesibles para los usuarios.

RNF-03. Seguridad: Los datos almacenados estarán protegidos y solo accesibles para los usuarios registrados en el sistema.

RNF-04. Disponibilidad: El sistema debe ser accesible en todo momento a través de un navegador web, sin necesidad de instalación de software adicional.

Análisis de requerimientos

Priorización

A continuación se encuentra la lista de funcionalidades del sistema:

- | | |
|---------------------------------------|----------------------------------|
| 1. Gestión de grupos. | 10. Historial de resultados. |
| 2. Gestión de usuarios. | 11. Notificaciones. |
| 3. Gestión de evaluaciones. | 12. Realización de evaluaciones. |
| 4. Banco de preguntas. | 13. Seguimiento académico. |
| 5. Reactivación de evaluación. | 14. Preguntas con multimedia. |
| 6. Autenticación de usuarios. | 15. Diseño responsive. |
| 7. Gestión de cuenta docente. | 16. Interfaz simple. |
| 8. Exportación de calificaciones. | 17. Seguridad. |
| 9. Generación de reportes académicos. | 18. Alta disponibilidad. |

Para la cual aplicamos el método MoSCoW:

Must Have <i>Esenciales para la viabilidad del producto</i>	Should Have <i>Importantes, pero con menor prioridad</i>	Could Have <i>Deseables, pero no esenciales</i>	Won't Have <i>Se pueden eliminar</i>
Gestión de grupos.	Reactivación de evaluación.	Banco de preguntas.	Notificaciones.
Gestión de usuarios.	Exportación de calificaciones.	Gestión de cuenta docente.	Seguimiento académico.
Gestión de evaluaciones.	Generación de reportes académicos.	Preguntas con multimedia.	
Autenticación de usuarios.	Interfaz simple.	Diseño responsive.	
Historial de resultados.			
Realización de evaluaciones.			
Seguridad.			
Alta disponibilidad.			

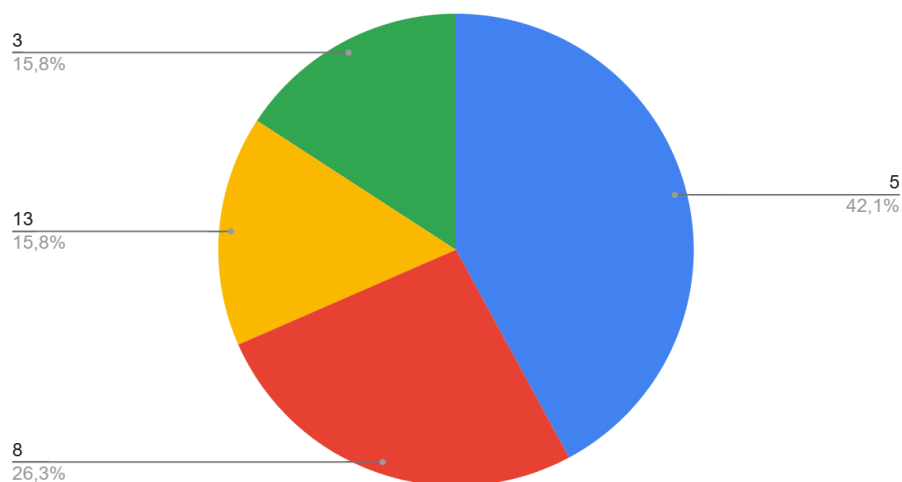
Estimación de esfuerzo

A continuación, se presentan los resultados del método Poker Planning llevado a cabo por el equipo para la estimación del esfuerzo utilizando la secuencia de Fibonacci (1, 2, 3, 5, 8, 13, 21, 34, 55, ...).

Funcionalidad	MoSCoW	Esfuerzo	Prioridad alta por protección de datos. Incluye autenticación, cifrado, etc.
Gestión de grupos (RF-01)	Must	5	Es esencial para organizar a los usuarios y gestionar permisos de acceso. Se requiere autenticación y estructura clara de datos.
Gestión de usuarios (RF-02, RF-04)	Must	8	Necesario para registrar, modificar y eliminar usuarios con distintos roles.
Asignación de materias y ligados a docentes (RF-03)	Must	5	Fundamental para la administración de cursos y profesores. Implica validaciones y reglas de negocio.
Gestión de evaluaciones (RF-05, RF-06, RF-07)	Must	13	Manejo complejo de evaluaciones, incluyendo creación, asignación y revisión. Necesita validaciones estrictas.
Banco de preguntas (RF-08)	Could	8	Facilita la reutilización de preguntas en diferentes evaluaciones. Agrega valor pero no es crítico para la operación inicial.
Reactivación de evaluación (RF-09)	Should	3	Permite que los estudiantes reanuden una evaluación en caso de interrupción, mejorando la experiencia de usuario.
Autenticación de usuarios (RF-10)	Must	8	Vital para la seguridad del sistema. Debe incluir autenticación robusta con cifrado y gestión de sesiones.
Gestión de cuenta docente (RF-10)	Could	5	Permite a los docentes gestionar su información personal y preferencias, mejorando la usabilidad.
Exportación de calificaciones (RF-11)	Should	5	Permite generar reportes descargables para seguimiento académico. Agrega valor, pero no es esencial.
Generación de reportes académicos (RF-12)	Should	5	Facilita el análisis de datos de desempeño. Necesario para la toma de decisiones, pero no urgente.
Historial de resultados (RF-14)	Must	5	Registro de calificaciones de los estudiantes. Importante para la trazabilidad académica.

Notificaciones (RF-15)	Won't	3	Útil para alertar a los usuarios sobre eventos, pero su implementación requiere una infraestructura adicional.
Realización de evaluaciones (RF-13)	Must	13	Componente clave del sistema. Debe manejar preguntas, respuestas, temporización y envíos seguros.
Seguimiento académico (RF-16)	Won't	5	Beneficioso para evaluar el progreso de los estudiantes, pero puede implementarse en fases posteriores.
Preguntas con multimedia (RF-17)	Could	8	Mejora la experiencia del estudiante, pero su desarrollo implica mayor esfuerzo técnico.
Diseño responsive (RNF-01)	Could	5	Importante para la accesibilidad en dispositivos móviles, pero no crítico para la funcionalidad principal.
Interfaz simple (RNF-02)	Should	3	Prioridad alta por protección de datos. Incluye autenticación, cifrado, etc.
Seguridad (RNF-03)	Must	8	Prioridad alta por protección de datos. Incluye autenticación, cifrado, etc.
Alta disponibilidad (RNF-04)	Must	13	Prioridad alta por protección de datos. Incluye autenticación, cifrado, etc.

Esfuerzo (Fibonacci)



El gráfico de pastel representa la distribución del esfuerzo estimado en la escala de Fibonacci (3, 5, 8 y 13). Se observa que el valor más frecuente (moda) es 5, con una frecuencia relativa del

42.1%, lo que indica que la mayoría de las funcionalidades tienen un nivel de esfuerzo intermedio. El rango de los datos es de 10 (13 - 3), lo que muestra una dispersión considerable en los valores de esfuerzo. La distribución es asimétrica, ya que hay una mayor concentración en los valores 5 y 8, mientras que los valores extremos (3 y 13) aparecen con menor frecuencia (15.8% cada uno). Esto sugiere que la mayoría de las funcionalidades presentan una dificultad moderada, pero existen algunas tareas con una mayor carga de trabajo. El uso de la secuencia de Fibonacci en la estimación implica un crecimiento no lineal del esfuerzo, lo que significa que pasar de un valor a otro no representa un incremento uniforme, sino un aumento progresivo de la complejidad. Por ejemplo, una funcionalidad con esfuerzo 13 no es solo un 62.5% más difícil que una de 8, sino que probablemente involucra factores adicionales como validaciones más estrictas, mayor integración entre módulos o procesamiento de datos más intensivo. En términos de planificación, la concentración de esfuerzos en valores bajos y medios sugiere que el proyecto tiene una carga de trabajo manejable en la mayoría de las funcionalidades, pero debe prestarse especial atención a aquellas con esfuerzo 13, ya que podrían representar cuellos de botella en el desarrollo.

Análisis de gestión del desarrollo

Tiempo

El desarrollo del sistema utilizará Python con Django para el backend y Vue.js para el frontend, con la siguiente planificación:

Fase	Tareas principales	Duración estimada
Análisis y Planificación	Definición de requerimientos, modelado de datos, arquitectura del sistema	1 semana
Diseño	Creación de wireframes, diseño de UI/UX	1 semana
Desarrollo Backend (Django)	Implementación de API REST, base de datos PostgreSQL, autenticación y seguridad	3 semanas
Desarrollo Frontend (Vue.js)	Implementación de interfaz de usuario y consumo de API	3 semanas
Pruebas y Ajustes	Pruebas unitarias, de integración y usabilidad	1 semana
Despliegue y Documentación	Implementación en servidor (AWS/GCP), documentación técnica y manual de usuario	1 semana

Costos

Los siguientes son los costos relacionados con el equipo de desarrollo que estará implicado en el proyecto:

Rol	Cantidad	Sueldo mensual (COP)	Total (2 meses) (COP)
Desarrollador Backend (Django)	1	\$2.685.791	\$5.371.582
Desarrollador Frontend (Vue.js)	1	\$2.685.791	\$5.371.582
Diseñador UX/UI	1	\$3.278.000	\$6.556.000
Tester QA	1	\$2.592.862	\$5.185.724
Arquitecto de Software	1	\$5.000.000	\$10.000.000

(Consultor)			
Total equipo			\$32.484.888

Adicionalmente, en la siguiente tabla se pueden ver otros costos directos relacionados con el desarrollo del proyecto:

Concepto	Costo Mensual (COP)	Total (2 meses) (COP)
Hosting en la nube (AWS, Azure, GCP)	\$600.000	\$1.200.000
Base de datos (PostgreSQL)	\$0	\$0
Licencias de software	\$800.000	\$1.600.000
Servicios de notificaciones	\$280.000	\$560.000
Total infraestructura		\$3.360.000

Producto mínimo viable (MVP)

El Sistema de Gestión de Exámenes en Línea está diseñado para cubrir las necesidades básicas de tres roles principales: Administradores, Docentes y Estudiantes. Las funcionalidades clave que se incluirán en el MVP son:

- **Gestión de Usuarios:**
Los administradores podrán crear, editar y eliminar usuarios, asignándoles un rol (Administrador, Docente, Estudiante) y credenciales de acceso. Los estudiantes se asociarán a un único grado y grupo.
- **Gestión de Grupos y Materias:**
Los administradores podrán crear y gestionar grupos de estudiantes, y los docentes se asignarán a materias y grados específicos para crear y administrar exámenes.
- **Creación y Gestión de Exámenes:**
Los docentes podrán crear exámenes con preguntas de opción múltiple, configurar fechas de entrega y asignar los exámenes a grupos específicos.
- **Realización de Exámenes:**
Los estudiantes podrán acceder a los exámenes a través de una interfaz en línea básica para la aplicación de exámenes.

- Seguridad Básica:

El sistema contará con autenticación de usuarios y cifrado básico para proteger la información.

Ahora, para garantizar que el MVP sea viable y se entregue a tiempo, algunas características deseables pero no esenciales se dejarán para futuras iteraciones del proyecto:

- Reportes:

Administradores y docentes podrán generar reportes de rendimiento académico por grupo, materia y estudiante, y exportar las calificaciones en formato Excel.

- Calificación Automática:

El sistema corregirá los exámenes con base en las respuestas correctas configuradas por el docente, y los estudiantes podrán ver sus resultados una vez finalizada la evaluación.

- Notificaciones Automáticas:

No se incluirán notificaciones automáticas (como recordatorios de exámenes o resultados) en esta versión. Tampoco habrá integración con plataformas externas como WhatsApp o correo electrónico.

- Seguimiento Académico Avanzado:

No se incluirán herramientas avanzadas para visualizar el progreso académico de los estudiantes, como gráficos o análisis comparativos.

- Preguntas con Multimedia:

No se permitirá la creación de preguntas con contenido multimedia (imágenes, videos o audios) en esta versión.

- Reactivación de Evaluaciones:

No se permitirá volver a habilitar una evaluación después de haber sido aplicada.

- Integración con Sistemas Externos:

No habrá integración con plataformas de mensajería (WhatsApp, correo electrónico) ni con otros sistemas de gestión escolar.

- Optimización para Tablets:

No se incluirá una optimización específica para tabletas en esta versión.

Diseño y Arquitectura

Para este proyecto se está empleando una arquitectura cliente-servidor, ya que estamos usando para el Frontend vue.js el cual es el cliente que se ejecuta en el navegador del usuario, para el Backend estamos usando Django que es el servidor que maneja la lógica del proyecto procesa las solicitudes y se comunica con la base de datos. Es cliente-servidor ya que el Frontend y el Backend son aplicaciones separadas y estas se comunican por APIs REST, esto hace que ambas puedan evolucionar por separado.

Con la comunicación mediante APIs REST el Frontend realiza solicitudes HTTP al Backend y este responde con datos en formato json así que estamos utilizando una separación por capas, ya que estamos usando Django entonces el patrón de este es el MTV (Model - Template - Vista)

Elección de esta Arquitectura

Principalmente elegimos este sistema por capas de Frontend con Vue.js, Backend con Django y base de datos porque creemos que es lo que mejor se ajustaba a la decisión de el tema del proyecto que tomamos y porque era más familiar para los integrantes del grupo sobre todo el trabajo con Django que ya habíamos trabajado con él y en parte por recomendación del profesor optamos también por Vue.js ya que aunque no lo habíamos trabajado pensamos que era una buena idea aprenderlo y emplearlo en este proyecto.

En cuanto a la escalabilidad sabemos que tanto el Frontend y el Backend tienen una buena escalabilidad ya que Vue.js es ligera y eficiente, lo que lo hace rápido en términos de rendimiento que es una parte esencial de los que necesitamos para este sistema de Exámenes además hemos visto que es medianamente fácil de integrar y Django es aún mejor la escalabilidad ya que es muy robusto y está diseñado para manejar aplicaciones de alto tráfico además de que Django tiene características de seguridad integradas.

Diseño Base de Datos

Para este proyecto se empleó una base de datos tipo SQL ya que para el sistema de calificaciones y lo que requiere sean grupos, profesores, administradores, estudiantes, etc. Requiere de una organización que a nuestro parecer es más fácil organizar con tablas y en un esquema bien definido, en cambio una base de datos NoSQL nos sería más difícil a nuestro parecer organizar toda esta información.

Explicando las decisiones tomadas en la base de datos, se crearon 15 tablas entre estas el grado, la asignatura, el grupo, el usuario, docente, estudiante, etc. Para muchas de estas se utilizaron llaves primarias utilizando un Id, que podía ser SERIAL o INT esto se hace para poder identificar claramente a qué grupo, usuario, estudiante nos referimos.

Al principio del código se eliminan algunos esquemas si es que ya existen, así como los objetos dependientes, después se crea el esquema sisexam, luego se empiezan a crear las tablas con sus respectivas relaciones para que el sistema funcione de manera adecuada por ejemplo grupo con grado y docente con usuario.

Las llaves Foráneas son empleadas para referenciar las claves primarias de otras tablas, así estableciendo relaciones entre tablas y así evitar confusiones y problemas (datos huérfanos o inconsistentes).

Para la base de datos se implementó la primera la segunda y la tercera forma normal, la primera es que cada columna contenga valores atómicos indivisibles y que cada fila sea única, la segunda forma normal, que todos los atributos no clave dependan completamente de la clave primaria. y la tercera forma normal, que no haya dependencias transitivas (es decir, los atributos no clave no deben depender de otros atributos no clave)

Referencias

Ministerio de Educación Nacional. (2015). *Decreto 1075 de 2015, Sector Educación*.

Ministerio de Educación Nacional. (2022). *Plan Estratégico Institucional (PEI) 2022-2026*.

Indeed. (s.f.). Salarios de desarrollador junior. Indeed. Recuperado el 31 de enero de 2025, de <https://co.indeed.com/career/desarrollador-junior/salaries>