

Gestión para la generación automática de exámenes

Requisitos funcionales, no funcionales y de entorno

Semigrupo: Aracelys

Equipo #4

C311

Kelen Alfaro García

Adriana Boué García

Carlos Mazorra Matos

Darián Santamarina Hernández

Dario Alfonso Urrutia

Requisitos Funcionales

1. Gestión de preguntas de examen:

- Alta, modificación, clasificación y eliminación de preguntas.
- Clasificación por tema, subtema, tipo de pregunta (opción múltiple, V/F, ensayo) y nivel de dificultad.
- Control para evitar duplicados.

2. Gestión de exámenes:

- Generación automática de exámenes en base a criterios (dificultad, tipo de pregunta, cobertura de temas).
- Revisión, validación y aprobación de exámenes por parte del jefe de asignatura.
- Posibilidad de generar otro examen si el revisor lo considera necesario.
- Parametrización de los criterios de generación y almacenamiento junto al examen.

3. Gestión de roles y usuarios:

- Profesores: creación de preguntas, generación de exámenes, validación/revisión de exámenes.
- Estudiantes: acceso a exámenes asignados, registro de respuestas, consulta de calificaciones, solicitud de recalificación.
- Administrador: control de duplicados y asignación de exámenes.

4. Gestión académica:

- Administración de asignaturas con su programa de estudios y temas.

- Gestión de profesores (identificador, nombre, especialidad, asignaturas autorizadas).
- Gestión de estudiantes (identificador, nombre, edad, curso, asignaturas matriculadas).

5. *Recalificación:*

- Solicitud de recalificación por parte del estudiante.
- Asignación de profesor encargado de la recalificación.
- Registro en base de datos de la nota antes y después de la recalificación.

6. *Consultas y reportes (tablas + gráficos):*

- Listado de exámenes generados por asignatura (con creador, fecha, parámetros).
- Preguntas más utilizadas en exámenes finales, clasificadas por dificultad y tema.
- Exámenes validados por un profesor revisor (con fecha y observaciones).
- Desempeño de estudiantes en un examen, comparando aciertos por dificultad.
- Correlación dificultad ↔ rendimiento, con top 10 preguntas más reprobadas.
- Comparación de exámenes por asignatura (distribución de preguntas y equilibrio).
- Registro de profesores que revisaron exámenes en los últimos dos semestres.
- Exportación de resultados a PDF.
- Ordenamiento de columnas en todas las consultas.

Requisitos No Funcionales

1. *Calidad de software:*

- Buenas prácticas de programación (código limpio).
- Implementar al menos dos patrones de diseño.
- Arquitectura desacoplada, extensible y mantenible.
- Pruebas unitarias tanto en back-end como en front-end.

2. *Usabilidad:*

- Interfaz web amigable, con consultas en tablas y gráficos interactivos.
- Posibilidad de ordenar y filtrar resultados.

3. *Transparencia y trazabilidad:*

- Registro histórico de creación, validación, calificación y recalificación de exámenes.

- Control de calidad en la generación automática de exámenes.
- 4. *Rendimiento:*
 - Capacidad de generar reportes estadísticos en tiempos aceptables.
 - Manejo eficiente de la base de datos para consultas complejas.
- 5. *Portabilidad / Multiplataforma:*
 - El sistema debe poder ejecutarse en diferentes sistemas operativos y navegadores.
- 6. *Confiabilidad:*
 - Garantizar el almacenamiento seguro de respuestas de estudiantes.
 - Integridad de datos frente a operaciones simultáneas (profesores/estudiantes).

Requisitos de Entorno

1. *Control de versiones:*
 - Uso de GitHub.
2. *Planificación:*
 - Uso de una herramienta CASE (GitHub Projects).
3. *Plataforma de ejecución:*
 - Aplicación web accesible desde navegadores modernos (Chrome, Firefox, Edge, etc.).
 - Multiplataforma: Windows, Linux, MacOS.
4. *Base de datos:*
 - Motor relacional (PostgreSQL) para garantizar integridad y consultas avanzadas.
 - ORM: Prisma
5. *Lenguajes y frameworks:*
 - Back-end: TypeScript (Node.js), Nest.js.
 - Front-end: TypeScript, Next.js.
 - Reportes PDF: Puppeteer.
6. *Pruebas:*
 - Frameworks de testing (Jest).