



## **Informe de Decisiones de Diseño**

José Antonio Serrano Contreras  
Mario Andres Andrade Muñoz  
Paradigmas de Programación  
Profesor: Jerry Jesús Peña Morales

Abril 2025

## 1. Introducción

El sistema desarrollado aborda la necesidad de gestionar pruebas y preguntas en un entorno educativo, facilitando la interacción entre profesores y alumnos. La motivación principal es automatizar y simplificar el proceso de creación, administración y resolución de evaluaciones.

### Objetivos

- Implementar un sistema de gestión de pruebas y preguntas
- Facilitar la creación y modificación de preguntas por parte del profesor
- Permitir a los alumnos responder pruebas de manera interactiva
- Mantener un registro organizado de preguntas y pruebas

## 2. Descripción de la Solución

### 2.1 Modelo de Clases

#### Clase Pregunta

*Objetivo:* Representar una pregunta individual con sus características.

*Funcionalidad:*

- Sistema de ID único usando contador estático
- Niveles taxonómicos para clasificación pedagógica
- Validación temporal de preguntas (vigencia de 2 años)
- Tipos de pregunta predefinidos

#### Clase Prueba

*Objetivo:* Agrupar preguntas en evaluaciones coherentes.

*Funcionalidad:*

- Gestión dinámica de preguntas
- Cálculo automático de tiempo total

#### Gestores (GestorPreguntas y GestorPruebas)

*Objetivo:* Administrar colecciones de preguntas y pruebas.

*Funcionalidad:*

- Manejo de memoria dinámica
- Operaciones CRUD
- Persistencia de datos mediante archivos

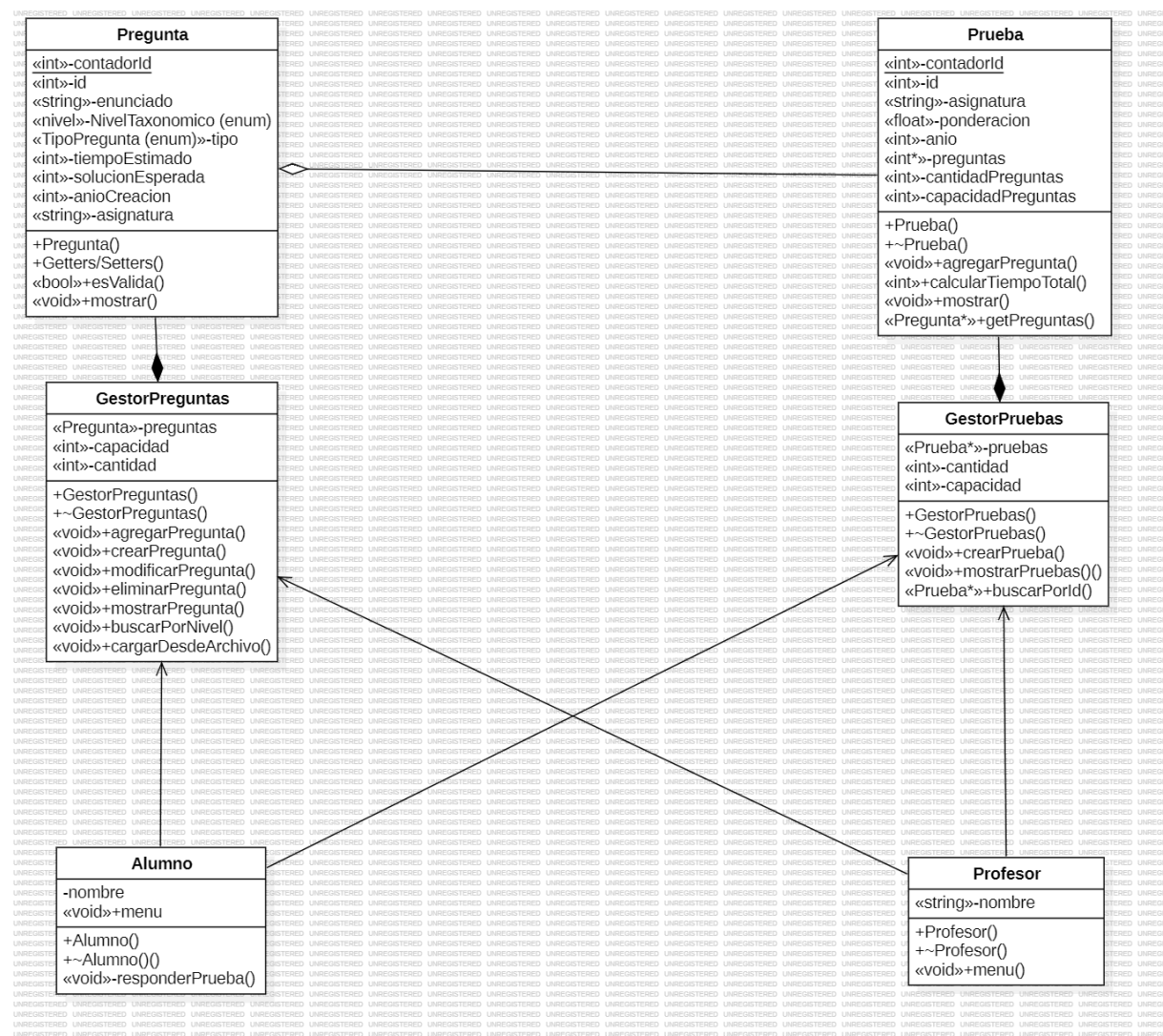
#### Clases de Usuario (Profesor y Alumno)

*Objetivo:* Definir interfaces específicas según el rol.

*Funcionalidad:*

- Profesor: Gestión completa del sistema
- Alumno: Visualización y respuesta de pruebas

## 2.2 Modelo de Clases



Esta representación visual muestra que:

- Los gestores son los dueños de sus respectivos objetos (composición)
- Las pruebas referencian preguntas pero no las poseen (agregación)
- Profesor y Alumno utilizan los gestores pero no los poseen (asociación)

## 2.3 Decisiones de Diseño

### Funcionalidad del Sistema:

El sistema consiste en la gestión de evaluaciones académicas. Divide sus funcionalidades en roles de usuario: Profesor y Alumno.

Para el profesor; permite crear, modificar y eliminar preguntas; crear pruebas seleccionando preguntas existentes; cargar preguntas desde archivos; ver todas las pruebas creadas y buscar preguntas por nivel taxonómico.

Para el alumno; permite ver las pruebas disponibles; responder pruebas y recibir retroalimentación inmediata sobre sus respuestas.

Las preguntas contienen pueden pertenecer a diferentes niveles taxonómicos (Recordar, Entender, Aplicar, etc.), tipos de preguntas (Verdadero/Falso, Respuesta Corta, Selección

Múltiple), asignatura, poseen un sistema de validez temporal (2 años desde su creación) y un tiempo estimado para responder.

Las pruebas son conformadas por diferentes preguntas, realizan un cálculo automático de tiempo total, se organizan por asignatura y año.

El sistema posee identificadores únicos para preguntas y pruebas, una interfaz de usuario basada en menús por consola, validación de entradas, y sistema de carga masiva de preguntas desde archivos.

#### Separación de Responsabilidades:

El diseño del sistema posee una separación de responsabilidades bien definida, permitiendo una estructura clara y mantenible. Se implementan gestores especializados para manejar tanto las preguntas como las pruebas, encapsulando la lógica de almacenamiento, carga, búsqueda y filtrado dentro de clases dedicadas. Además, la diferenciación de roles de usuario permite que profesores y alumnos interactúen con el sistema de manera distinta, cada uno con un menú específico que restringe o habilita funcionalidades según su perfil.

#### Gestión de memoria

El uso de arreglos dinámicos permite una mayor flexibilidad al almacenar colecciones de variables de preguntas y pruebas sin desperdiciar recursos. Para evitar fugas de memoria, se ha implementado un control riguroso de los recursos mediante destructores, garantizando que cada objeto libere correctamente la memoria reservada al finalizar su uso.

Además, al trabajar con arreglos dinámicos con punteros, fue necesario implementar correctamente constructores de copia y operadores de asignación para garantizar una gestión segura y eficiente de los recursos. Su inclusión permite que los objetos se copien y asignen de forma controlada, evitando errores como fugas de memoria, corrupción de datos o accesos inválidos.

#### Agregación de preguntas en pruebas

Se decidió asociar las preguntas a las pruebas haciendo referencia a ellas en lugar de contenerlas. Esto para no duplicar objetos y mantener la gestión de estos en la clase gestora correspondiente.

### **3. Conclusión**

El sistema implementado cumple con los objetivos propuestos, proporcionando una plataforma funcional para la gestión de pruebas, permitiendo la interacción efectiva entre profesores y alumnos con las pruebas y preguntas, y manteniendo la integridad y organización de los datos.

La programación orientada a objetos ha sido ideal para el desarrollo de este sistema, ya que permite una organización modular y escalable del código. Gracias a esta estructura, se facilita la adición de nuevos tipos de preguntas en el futuro sin necesidad de reescribir grandes secciones del programa. Asimismo, el mantenimiento del sistema se simplifica al contar con clases bien definidas, cada una con responsabilidades específicas, implementando una correcta separación de responsabilidades.

El sistema actual es funcional y refleja una implementación sólida de los principios de la programación orientada a objetos, lo que se traduce en un producto confiable y fácil de mantener. No obstante, existen áreas de mejora que podrían enriquecer significativamente la experiencia del usuario y la utilidad del sistema: implementación de un sistema de calificación más profundo, mejoras en la interfaz de usuario para facilitar la navegación, implementación de herencia en clases de usuario. Entre otras.

El desarrollo de este proyecto tuvo grandes dificultades y oportunidades de aprendizaje, especialmente en lo referente a este paradigma de programación.